1968

A sensitivity analysis of the policy-iteration technique for solving Markovian decision problems

William L. Nutter
Lehigh University

A SENSITIVITY ANALYSIS OF THE POLICY-ITERATION TECHNIQUE FOR SOLVING MARKOVIAN DECISION PROBLEMS

by

W. L. Nutter

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in Industrial Engineering

Lehigh University
1968

CERTIFICATE OF APPROVAL

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

30 April 1968
Date

John W. Adams
Professor in Charge

Head of the Department

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES (cont'd)

ABSTRACT

The policy-iteration technique is a method of selecting the optimal solution vector for a Markovian decision problem. This technique was analyzed to determine its sensitivity with respect to random errors in the stochastic elements in such problems. Two factors were chosen to reflect sensitivity as a function of the standard deviation of a random normal perturbation applied to the transition probabilities. The first factor is a relative frequency approximation to the probability of selecting a non-optimal policy vector and the second is the expected cost of such a selection expressed as a percentage of the optimal return per transition period. The sensitivity of the method, as reflected by those two parameters, was found to vary over a wide range. These variations are strictly dependent upon the structure of the problem being tested.

CHAPTER I

Introduction

The general area of interest in this paper is that of Markovian decision problems. Problems of this type comprise a subset of a more general class known as sequential decision problems. The characteristic of Markovian decision problems that distinguishes them from the more general class is a certain independence in the decision-making process at sequential states of the system. This is generally referred to as the Markov property of the system. More specifically our interest will be directed to a particular technique for solving Markovian decision problems. This method was developed by Ronald A. Howard for an Sc.D thesis at M.I.T. in 1958. Howard's method is an iterative technique similar to dynamic programming. Howard refers to the process as the policy-iteration technique.

The purpose of this thesis is to determine what can be said regarding the sensitivity of the policy-iteration method. In particular, we are concerned with the technique's sensitivity with respect to random errors in the probabilistic elements in Markovian decision problems. The basic approach adopted was to repeatedly impose random perturbations, with controlled parameters, upon the transition matrix of a given problem and attempt to measure the resultant sensitivity as the parameters of perturbation were systematically varied.

Markov Processes

We will present here a brief introduction to Howard's method, deferring until later a more detailed discussion. In order to present a reasonably clear picture of the application of the policy-iteration technique to the solution of Markovian decision problems we must first discuss Markov chains or Markov processes. Markov processes do not lend themselves to a brief one-sentence definition. Perhaps they can best be described by example, which we will give shortly. A Markov process is a mathematical structure characterized by the concepts of "states" and "state transitions". A "state" can be defined as a unique condition of the system which can be completely described by a specific set of system parameters. "State transitions" refer to a process wherein the condition of the system undergoes a change from one uniquely defined state to another. Transitions from one state to another are governed by an array of probabilities referred to as the transition matrix for the process.

A rather picturesque example of a Markov process, given by Howard[3], is that of a frog in a lily pond. From time to time the frog jumps from one lily pad to another. If each of the lily pads were assigned a number, then the state of the system would be defined by specifying the number of the pad currently occupied by the frog. A leap from one pad to another constitutes a state transition. Assuming that it is possible to do so, for the sake of our example, we would assign probabilities to each of the possible jumps. That is, we would construct a matrix whose entries constitute an exhaustive enumeration of the probabilities associated with all possible

transitions. The matrix would be N x N, where N is the number of lily pads in the pond. This is the transition matrix for the process.

There is one further restriction we must place on our system. The probability of a transition from pad i to pad j is dependent only upon i and j. That is, the probability of going from pad i to pad j is independent of how the frog arrived at pad i. This "Markov property" is the one that qualifies the Markov process as a special case of the more general class of stochastic processes.

Markovian Decision Problems

Now that we have described a Markov process we can proceed to the development of a Markovian decision process. To do this we need to introduce the concepts of alternatives and rewards. Suppose in our example two observers decide to gamble on the frog's jumps by placing bets on which pad he will jump next. To simplify the betting we could agree on a matrix of rewards whose entries give the payoffs for all possible jumps. Thus there would be a one-to-one correspondent between the entries in the transition and reward matrices. Depending on how we agree to play the game some of the entries in the reward matrix may or may not be negative. Suppose further that we have observed that we can affect the probabilities of the frog's jumps by various actions on our part such as clapping our hands or throwing rocks in various parts of the pond. Thus corresponding to a given state of the system there might be a number of alternatives which we could adopt to affect the transition probabilities. To keep the game fair we would have to agree on different rewards

corresponding to each action or alternative that we adopt for a given state.

We have now described a Markovian decision process although admittedly a rather fanciful one. To summarize we can say that a Markovian decision process is a sequential decision problem in which the Markov property holds and further

(1) corresponding to each state of the system there are a number of possible courses of action and

(2) to each alternative adopted in a given state there exists a corresponding transition probability and a corresponding reward.

The question naturally arises as to how we should play the game to optimize our expected reward or minimize our losses. A solution to the decision process is obtained by selecting the alternative to be adopted in each of the possible states. This set of alternatives is referred to as a policy vector. The optimum policy vector is the set of alternatives which maximizes the long range expected reward. Howard's policy-iteration technique is a method of selecting the optimum policy vector.

Purpose

The variables affecting the selection of the optimum policy are the transition and reward matrices. We would expect the rewards associated with the decision problem to be known with a fairly high degree of confidence. At least they will be known with a much higher degree of confidence than will the transition probabilities.

It would be interesting to observe how sensitive the policy-iteration technique is to errors in estimation of these probabilities. Specifically, the purpose of this thesis is to determine what can be said regarding the sensitivity of Howard's method to errors in the transition probabilities. The experimental procedure adopted to answer this question is discussed in detail in Chapter III.

Report Format

This paper consists of four chapters and an appendix. Chapter I includes an introduction to Markovian decision problems, a statement of the purpose of the thesis and a description of the format of the discussion. Chapter II contains a detailed discussion of Markovian decision problems and the policy-iteration technique for solving this type problem. Chapter III is devoted to a discussion of the experimental procedure used in making this study. Finally Chapter IV presents the results of this study along with the conclusions that may be drawn from these results. Supporting material is included in the appendix.

CHAPTER II

Markov Chain Theory

Our ultimate goal in this chapter is to develop a basic understanding of Howard's policy-iteration technique. To do this, we must start with a discussion of the theory of Markov chains. The development given here is basically an adaptation of the material presented by Howard[1] and Kemeny, Snell[5,6,7] et. al.

Since its introduction in 1907 by A. A. Markov, the theory of Markov chains has become well established and there are some excellent texts available on the subject. It is not our purpose here to delve into an extensive study of Markov chains. Rather we need only to establish a basis for discussing Markovian decision problems and subsequently Howard's technique for solving them.

We are interested in discrete-time processes with constant transition times. Continuous-time processes are those in which the time between transitions is a random variable. Requiring the transition to be constant is not unduly restrictive since this is quite often true in practical situations, e.g. inventory control problem with fixed ordering intervals.

Let us denote by p_{ij} the probability of a transition to state j given we are now in state i and observe that

$$\sum_{j=1}^{N} p_{ij} = 1$$

where N is the total number of states in the system. We will refer to the transition matrix for the process as P which is

given by

$$P = [P_{ij}] = \begin{bmatrix} P_{11} & P_{12} & \cdots\cdots & P_{1N} \\ P_{21} & P_{22} & \cdots\cdots & P_{2N} \\ \vdots & & & \vdots \\ P_{N1} & P_{N2} & \cdots\cdots & P_{NN} \end{bmatrix}$$

Notice that P is N x N and the rows sum to 1. This matrix gives a complete description of the Markov process.

Through appropriate manipulations of the transition matrix we can answer all reasonable questions about the process. For instance we can determine the probability that the system is in any specific state after any specified number of transitions if we know the starting state. In many cases the starting state need not even be known if the number of transitions is large. In line with this thinking, we will define a state probability $x_i(n)$ as the probability that the system will be in state i after n transitions given the starting state, i.e., the state at n = 0. Similarly we define a row vector X(n) with components $x_i(n)$. Thus the components of X(n) would be the respective probabilities of being in the states 1 through N after n transitions given the starting state. Since the process must be in some state after n transitions it follows that

$$\sum_{i=1}^{N} x_i(n) = 1$$

A useful recurrence relation can be developed as follows. The probability of being in state j after n+1 transitions, $x_j(n+1)$,

is given by the product of the probability of being in state i after n transitions times the probability of a transition from state i to state j summed over all i. In equation form this is

$$x_j(n+1) = \sum_{i=1}^{N} x_i(n)p_{ij} \quad n = 0,1,2,\ldots \qquad (2.1)$$

Simple matrix theory then enables us to write

$$X(n+1) = X(n)P \qquad (2.2)$$

By recursion we see that

$$X(1) = X(0)P$$

$$X(2) = X(1)P = X(0)P^2$$

$$X(3) = X(2)P = X(0)P^3$$

and in general

$$X(n) = X(0)\ P^n \qquad n = 0,1,2,\ldots \qquad (2.3)$$

Therefore we can find the probability that the system is in each of its states after n transitions by postmultiplying the initial-state probability vector X(0) by the n^{th} power of the transition matrix P. Normally, the starting state will be specified (i.e., non-probabilistic) so that X(0) will have a one in one position and zeroes elsewhere.

It is interesting to observe the behavior of the state probability vector as n becomes large. Many Markov processes exhibit the property that the state probability vector approaches a constant as n increases and further that the value of this constant vector is independent of the starting state. Howard defines an ergodic Markov process as one whose limiting state probability distribution is independent of starting conditions. We will deal

exclusively with ergodic processes. This will be discussed further in Chapter III.

For ergodic processes let us define x_i as the probability that the system is in state i after many transitions. The row vector X with components x_i is the set of probabilities commonly referred to as the steady-state probabilities for the process. It follows that

$$X = XP \tag{2.4}$$

and

$$\sum_{i=1}^{N} x_i = 1 \tag{2.5}$$

Combining equations 2.4 and 2.5 we can find the steady-state probabilities directly from the transition matrix.

Markov Process with Rewards

Suppose our N-state Markov process generates a sequence of rewards r_{ij} as it makes transitions from state to state. The set of rewards for the process can be described by the reward matrix R whose entries are the elements r_{ij}. Thus there is a one-to-one correspondence between the entries of R and P.

Let us define $v_i(n)$ as the expected total reward to be earned in the next n transitions given the system is now in state i.

If we define q_i as the immediate expected reward for state i, that is, the expected reward resulting from the next transition given we are in state i, we can write

$$q_i = \sum_{j=1}^{N} p_{ij}\, r_{ij} \qquad i=1,2,\ldots,N \tag{2.6}$$

Since the expected reward to be earned in the next n transitions consists of the reward for the first transition plus that for the n-1 remaining transitions we have the recurrence relation

$$v_i(n) = q_i + \sum_{j=1}^{N} p_{ij} c_j(n-1) \qquad i=1,2,\ldots,N \quad n=1,2,\ldots \qquad (2.7)$$

Equation 2.7 may be written in vector form as

$$V(n) = Q + P\,V(n-1) \qquad n=1,2,3,\ldots \qquad (2.8)$$

where V(n) and Q are column vectors with N components.

It would be useful to establish a method of measuring the average reward generated by our Markov process. To do this we will define the quantity g as the average expected gain per transition for the system. To calculate g we need only to sum the products of q_i and x_i over all i. Since x_i gives the probability that the system is in state i and q_i gives the immediate expected reward earned in the next transition from state i, g then is given by the product of x_i and q_i summed over all i which yields

$$g = \sum_{i=1}^{N} x_i q_i \qquad (2.9)$$

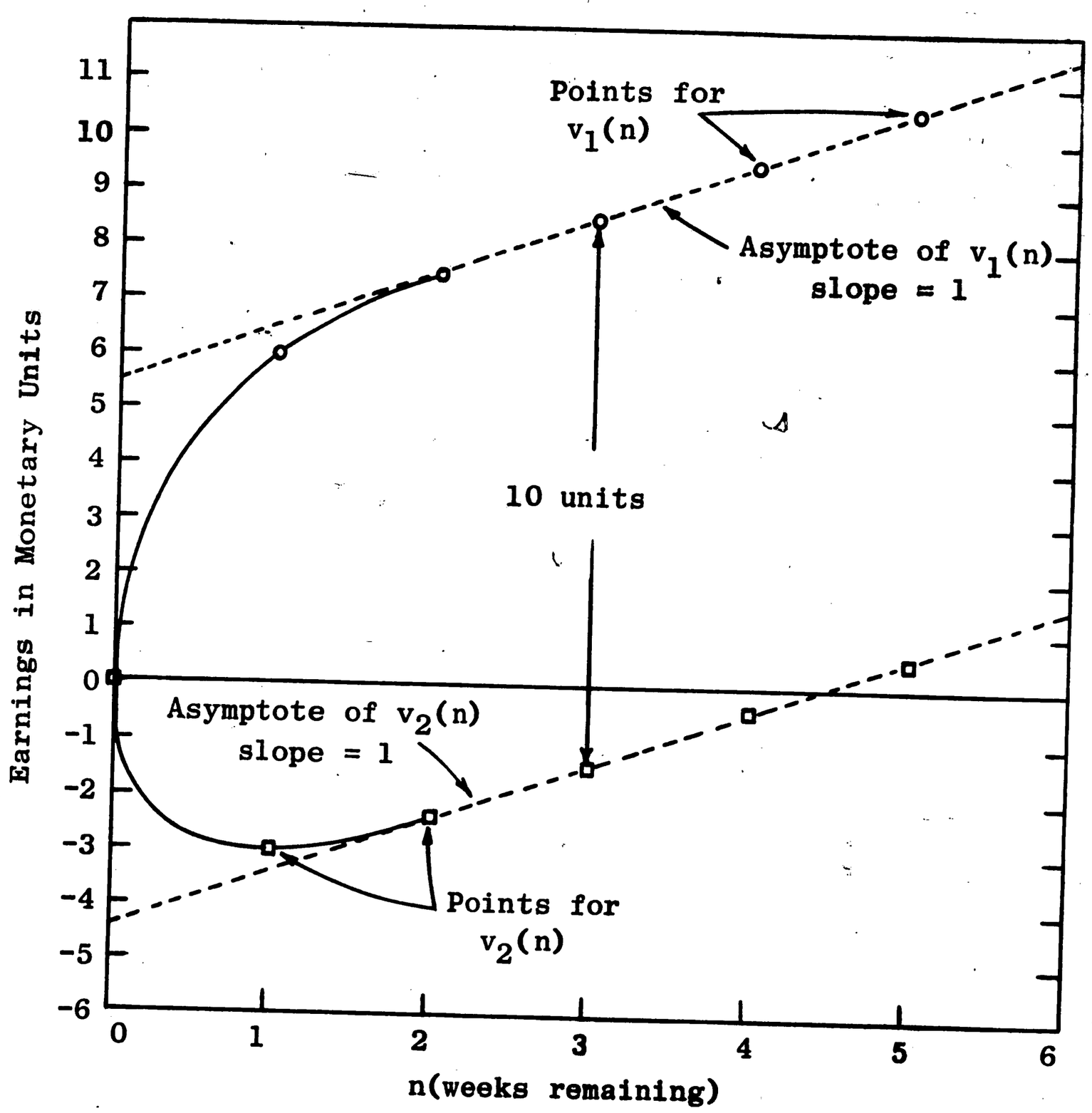
For non-ergodic processes we would have to define g differently since the gain would depend on the initial starting state. Since we are interested in ergodic processes, g will be independent of starting conditions. Now since g is constant we know that $v_i(n)$ must be a straight line function after the system has

reached steady-state conditions. Therefore we are justified in writing

$$v_i(n) = ng + v_i \qquad (2.10)$$

as an asymptotic representation for $v_i(n)$, where v_i denotes the asymptotic intercept of $v_i(n)$. Although 2.10 does not describe the transient response of the system, it is an exact representation of the steady-state response. Since we will ultimately be concerned with the behavior of the system for large n, equation 2.10 will be very useful.

The relationship of eq. 2.10 to the actual behavior of the system is illustrated by Figure 2.1. This graph depicts the total expected reward of a two-state problem discussed by Howard (the toymaker's problem). The solid lines give the actual expected reward as a function of n whereas the broken lines depict the asymptotic behavior of the process. Notice that the asymptotic and actual representations are coincident once steady-state has been reached. For this problem g is one. That is, the expected reward per transition is one unit. Notice also that the gain is independent of the starting state. The difference in the asymptotic intercepts reflect the desirability of starting in state 1 rather than state 2 so that the total expected reward is greater (by ten units) if the process starts in state 1 but the average expected reward per transition is the same regardless of the starting state.



TOYMAKER'S PROBLEM; TOTAL EXPECTED REWARD IN EACH STATE AS A FUNCTION OF WEEKS REMAINING

FIGURE 2.1

Markovian Decision Problems

The discussion of Markov processes with rewards was the logical basis from which to develop the structure of sequential decision problems of Markovian nature. In our discussion of this type problem we will use an example described by Howard.

Howard poses the problem of a toymaker who may be in either of two states. By definition, he is in state 1 if the toy he is currently producing is favorably accepted by the public and in state 2 if the opposite is true. Suppose that the probability that he will stay in state 1 at the end of a transition period (one week) is .5 and consequently the probability that he will make a transition to the undesirable state 2 is also .5. Further, suppose that the probability of goind from state 2 to state 1 at the end of the week is .4 and that of staying in state 2 is .6. Thus the transition matrix for the process is given by

$$P = [p_{ij}] = \begin{bmatrix} .5 & .5 \\ .4 & .6 \end{bmatrix}$$

The concept of rewards is introduced as follows. Suppose the toymaker earns a reward of 9 units when he has a successful toy in two successive weeks, i.e., the system makes a transition from state 1 to state 1. Thus $r_{11} = 9$. If the week has resulted in a transition from unsuccessful to unsuccessful (state 2 to state 2) then the toymaker loses 7 units or $r_{22} = -7$. Finally let $r_{21} = r_{12} = 3$. The reward matrix is therefore

$$R = [r_{ij}] = \begin{bmatrix} 9 & 3 \\ 3 & -7 \end{bmatrix}$$

Howard presents a complete analysis of this Markov process with rewards which will not be necessary for our purposes. The interested reader is referred to Howard's text[3]. Our purpose in presenting this example is to facilitate our understanding of sequential decision problems. To qualify our example as a decision problem we must now introduce the concept of alternatives.

Suppose the toymaker can govern his actions so as to modify the probabilities and rewards associated with the process. For example, when the toymaker has a successful toy, he may use advertising to decrease the probability of the toy falling from favor. Due to the costs of advertising the profits may be reduced but still greater than that resulting from a transition to an unfavorable toy. Naturally if the system still makes a transition to state 2, even after advertising was employed, the corresponding reward will also be reduced. Thus the toymaker has two alternatives in state 1. He may use no advertising or he may advertise. Denoting the alternatives by the superscripts 1 and 2 respectively, let us suppose that $r_{11}^{2} = 4$ and $r_{12}^{2} = 2$. From previous discussion we have $r_{11}^{1} = 9$ and $r_{12}^{2} = 3$. (If the reader refers to Howard's text he will note a discrepancy in Howard's discussion at this point. He allows r_{12}^{2} to be greater than r_{12}^{1}. One might argue that a transition from state 1 to state 2, when advertising, would mean

that the toy is less unfavorably received than when no advertising is employed. The fallacy in this reasoning is that it destroys the uniqueness of definition of the states).

There may also be various alternatives available in state 2. Increased research expenditure will enhance the probability of a favorable toy next week but will result in a correspondingly lower reward (due to the cost of research). We will refer to the alternative of no research as alternative 1 and to that of employing research as alternative 2. Let us suppose that $r_{21}^{2} = 1$ and $r_{22}^{2} = -19$. (Note the contrasting consistency of Howard's reasoning regarding r_{22}^{2} as compared to r_{12}^{2}.)

We mentioned that reason for introducing the alternatives of research and advertising was to enhance the probabilities of desirable transitions. In agreement with Howard let us suppose that

$$p_{1j}^{2} = [0.8 \quad 0.2]$$

and

$$p_{2j}^{2} = [0.7 \quad 0.3]$$

Recall that

$$p_{1j}^{1} = [0.5 \quad 0.5]$$

and

$$p_{2j}^{1} = [0.4 \quad 0.6]$$

Recall also that we defined a policy as the specification of the alternative to be pursued in each case. Thus in our example there

are four possible policies, namely, $\begin{bmatrix}1\\1\end{bmatrix}$, $\begin{bmatrix}1\\2\end{bmatrix}$, $\begin{bmatrix}2\\1\end{bmatrix}$ and $\begin{bmatrix}2\\2\end{bmatrix}$ where the entry in the i^{th} position of a policy vector denotes the number of the alternative to be adopted when in the i^{th} state. We now seek a method for determining which of the possible policies is optimum. We define the optimum policy as that one which maximizes our long range expected reward. Reflection will indicate that this is also the policy which maximizes g, the average expected gain per transition. Howard's policy-iteration technique provides us with a method of finding this optimal policy.

The Policy-Iteration Method

The policy-iteration method is an iterative process which basically permits us to solve for the gain and "relative values" of the system for a given policy and then find the optimum policy given those gains and "relative values" in a cyclic procedure which settles when the gain has been optimized. It involves repetitive solutions to sets of simultaneous equations and repetitively choosing the maximum of a set of calculated quantities. The process is more difficult to describe than it is to implement.

Recall the recurrence relation given by equations (2.7)

$$v_i(n) = q_i + \sum_{j=1}^{N} p_{ij} v_j(n-1) \quad i=1,2,\ldots N \quad n=1,2, \qquad (2.7)$$

Since we are concerned about the long run performance of our system we can use

$$v_i(n) = ng + v_i \qquad (2.10)$$

to substitute for $v_i(n)$ in equations 2.7 yielding

$$ng + v_i = q_i + \sum_{j=1}^{N} p_{ij} (n-1)g + v_j \quad i=1,2,\ldots,N \qquad (2.11)$$

Noting that $\sum_{j=1}^{N} p_{ij} = 1$, equations 2.11 become

$$g + v_i = q_i + \sum_{j=1}^{N} p_{ij} v_j \quad i=1,2,\ldots,N \qquad (2.12)$$

Howard notes that equations are unchanged if we add a constant to all v_i. This means that we cannot use equations 2.12 to solve for the v_i but fortunately we are not interested in absolute values. By arbitrarily setting $v_n = 0$ we can solve for v_i that only differ from the true v_i by a constant amount. Since we are interested in the differences between v_i, which are independent of absolute values, the v_i thus obtained are sufficient. It is these v_i that Howard calls "relative values." So we now have a method of obtaining the gain and relative values as a function of the policy. We will next formulate a method of obtaining the optimum policy as a function of the relative values. This will be a policy which has a higher gain than the original policy. (A proof of this statement is given by Howard).

Replacing n by n+1 in equations 2.7 we have

$$v_i(n+1) = q_i + \sum_{j=1}^{N} p_{ij}\, v_j(n) \qquad i=1,2,\ldots,N \quad n=1,2, \qquad (2.13)$$

We will use superscripts ^{k} to denote the q_i and p_{ij} corresponding to alternative k. That is, q_i^k and p_{ij}^k are the q_i and p_{ij} corresponding to adopting alternative k in state i. Now if we had followed an optimal policy up to stage n it follows that we could find the optimal policy at stage n+1 by maximizing

$$q_i^k + \sum_{j=1}^{N} p_{ij}^k\, v_j(n) \qquad (2.14)$$

over all k for each i. For large n we can substitute equation 2.10 to yield

$$q_i^k + \sum_{j=1}^{N} p_{ij}^k (ng+v_j) \qquad (2.15)$$

as the quantity to be maximized over all k in each state i. Since ng is independent of j in the summation we note that

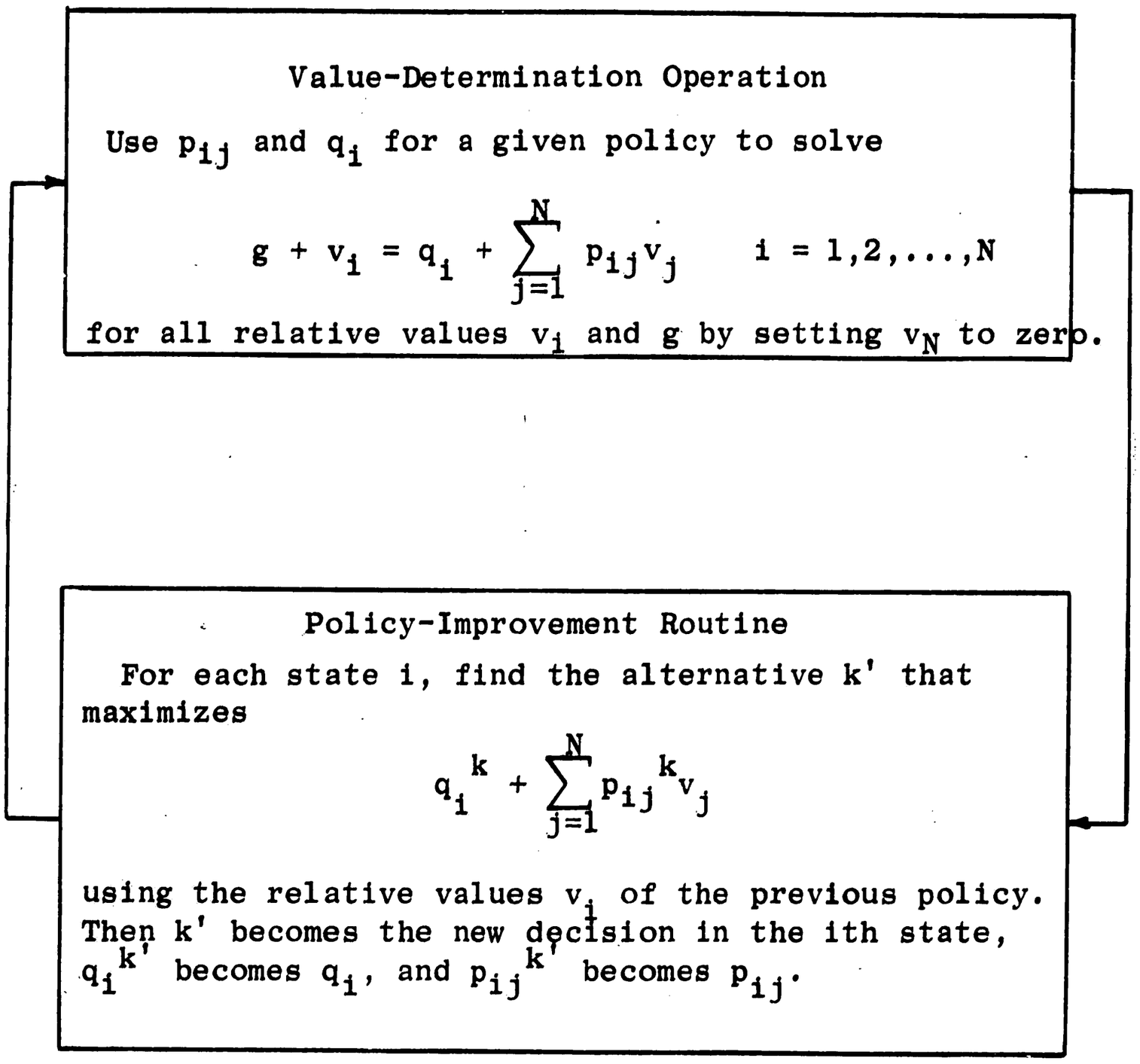
$$ng \sum_{j=1}^{N} p_{ij}^k = 1$$

so that 2.15 becomes

$$q_i^k + \sum_{j=1}^{N} p_{ij}^k v_j \qquad (2.16)$$

Thus we will maximize 2.16 over k at stage n+1 and further we will use the v_j calculated using equations 2.12. We now have established the iterative cycle as follows. For each state i find the alternative that maximizes 2.16 using the relative values determined from equations 2.12. Then use these alternatives to define the q_i and p_{ij} in equation 2.12 and re-solve for g and v_i. When the same policy is obtained twice in succession this becomes the optimal policy for the decision problem and the iterative process is terminated. The iteration cycle can be illustrated by the diagram shown in Figure 2.2.

Referring to Figure 2.2 we see that the upper box, the value-determination operation, yields the gain and relative values as a function of the policy. The policy-improvement routine yields the policy that increases the gain for a given set of v_i. We then re-enter the value-determination box with this policy and re-calculate g and



THE ITERATION CYCLE

FIGURE 2.2

v_i. Next we re-enter the lower box with these v_i and repeat the process.

The question of how to get started will naturally arise. Notice that if we start in the upper box we must specify a starting policy. If we have no particular preference as to the starting policy we can conveniently enter the lower box first with all $v_i = 0$. Thus we would maximize q_i over k. Recalling the definition of q_i, this means we will initially find the policy that maximizes immediate expected reward.

The interested reader is referred to Howard for a proof of the properties of the policy-iteration method.

CHAPTER III

Experimental Procedure

This chapter will be devoted entirely to describing the experimental procedure. Initially a FORTRAN program was written to implement the policy iteration method on the computer. This package then became the central framework around which the test was designed.

To simulate errors in estimation of the transition probabilities, a known problem was solved and solution data stored. Next the transition matrices were systematically perturbed and the problem resolved. The perturbations were applied by drawing numbers from a random normal distribution with controlled parameters. One hundred repetitions of this process were implemented for each value of perturbation parameters and comparisons drawn between these results and those obtained for the original un-perturbed problem.

To facilitate a detailed description of the experiment let us define the following parameters. Let

g_{ms} = the optimum gain of m^{th} perturbed system corresponding to a given value of s.

$$\Delta g_{ms} = g_o - g_{ms}$$

$$g_{cs} = \sum_{m=1}^{100} \Delta g_{ms}/g_o$$

v_{cs} = the number of non-optimal policy vectors selected in a sample of one-hundred taken for a given value of s.

v_m = variance of the gain taken over all possible policies for the un-perturbed problem (about the mean).

v_o = variance of the gain taken over all possible policies for the un-perturbed problem (about the optimum).

g_o = the gain of the original un-perturbed system operating under the optimal policy.

g_{os} = the gain of the original un-perturbed system operating under the second best policy, i.e., that policy nearest the optimum.

$\delta = g_o - g_{os}$

$$t^k_{ijs} = (p^k_{ij} + e_s) \Big/ \sum_{j=1}^{N} (p^k_{ij} + e_s) \qquad (3.1)$$

where:

p^k_{ij} = the probability of a transition from state i to state j corresponding to the adoption of alternative k.

e_s = a random number drawn from a normal population having mean zero and variance s^2.

t^k_{ijs} = the entries in the perturbed transition matrices.

Since each of the rows of the perturbed matrices must sum to one and all entries must be non-negative, the perturbations are subject to the following constraints.

(1) $t^k_{ijs} \geq 0$ for all i,j,k and s

(2) $\sum_{j=1}^{N} t^k_{ijs} = 1$ for all i,k and s

The perturbations were applied in the following manner. The IBM subroutine GAUSS yields random numbers normally distributed with controlled mean and standard deviation. Starting with p^1_{11} a random number with mean zero and variance s^2 is generated and added to p^1_{11} to form a temporary sum. This sum is checked to determine if it is negative. If the sum is negative, it is rejected and another random number generated and the sum re-formed. This process is repeated until the temporary sum is non-negative. This insures that constraint (1) will be satisfied. At this point the sum is stored as the t^1_{11s} entry in our transition matrix and we move to p^1_{12}. This procedure is repeated for p^1_{11} through p^1_{1N}. We next form the sum $\sum_{j=1}^{N} t^1_{1js}$ and then divide each t^1_{1js} from j = 1 to j = N by this sum. This ensures that constraint (2) will be satisfied. For i = 1, this procedure is repeated for k rows. The process is continued for all i from 1 to N.

The foregoing describes the method of perturbing the transition matrices. One-hundred such perturbations are imposed for each value of s and each of the one-hundred resulting problems are solved and solution data compared with that obtained for the original problem. For each problem, the selected policy vector is compared with the optimal vector for the un-perturbed problem. If the two vectors are unequal, v_{cs} is incremented by one. We refer to v_{cs} as the count of "non-optimal" policy vectors selected. We should briefly discuss the use of the terminology "non-optimal".

The policy vector selected for a given perturbed problem is certainly optimal for that problem. It is only non-optimal in the sense that it is a reflection of the effect of perturbing the sto-

chastic elements in the decision process. Since this is the effect we seek to measure it seems reasonable to refer to such vectors as non-optimal selections when they differ from the optimal policy vector for the un-perturbed system. Therefore for each value of s, v_{cs} represents the total of such non-optimal selections in a sample of one-hundred. We might view $v_{cs}/100$ as a relative frequency approximation for the probability of a non-optimal selection with a given estimation error on the transition probabilities.

For each of the one-hundred perturbed systems (for a given value of s) we also record Δg_{ms}. Δg_{ms} is the difference between g_o, the optimum gain of the original system, and the "optimum" gain of the m^{th} perturbed system (m ranges from 1 to 100). We then average Δg_{ms} over the one-hundred problems solved and obtain g_{cs} by dividing this average by g_o to form a percentage. Thus g_{cs} is the average percentage cost in gain resulting from the v_{cs} non-optimal policy selections. That is, it is the average cost in gain expressed as a percentage of the optimum gain of the system. Therefore, while v_{cs} is indicative of the probability of selecting a non-optimal policy, g_{cs} reflects the actual significance of such a selection.

This entire procedure is repeated for each value of s which ranges from .01 to .30 in increments of .01.

We might briefly examine the implications of the method of imposing the transition perturbations. Notice that a normalizing procedure is involved. This normalizing procedure virtually precludes the existence of a unity probability as any entry in a perturbed transition matrix. This means that the resultant processes

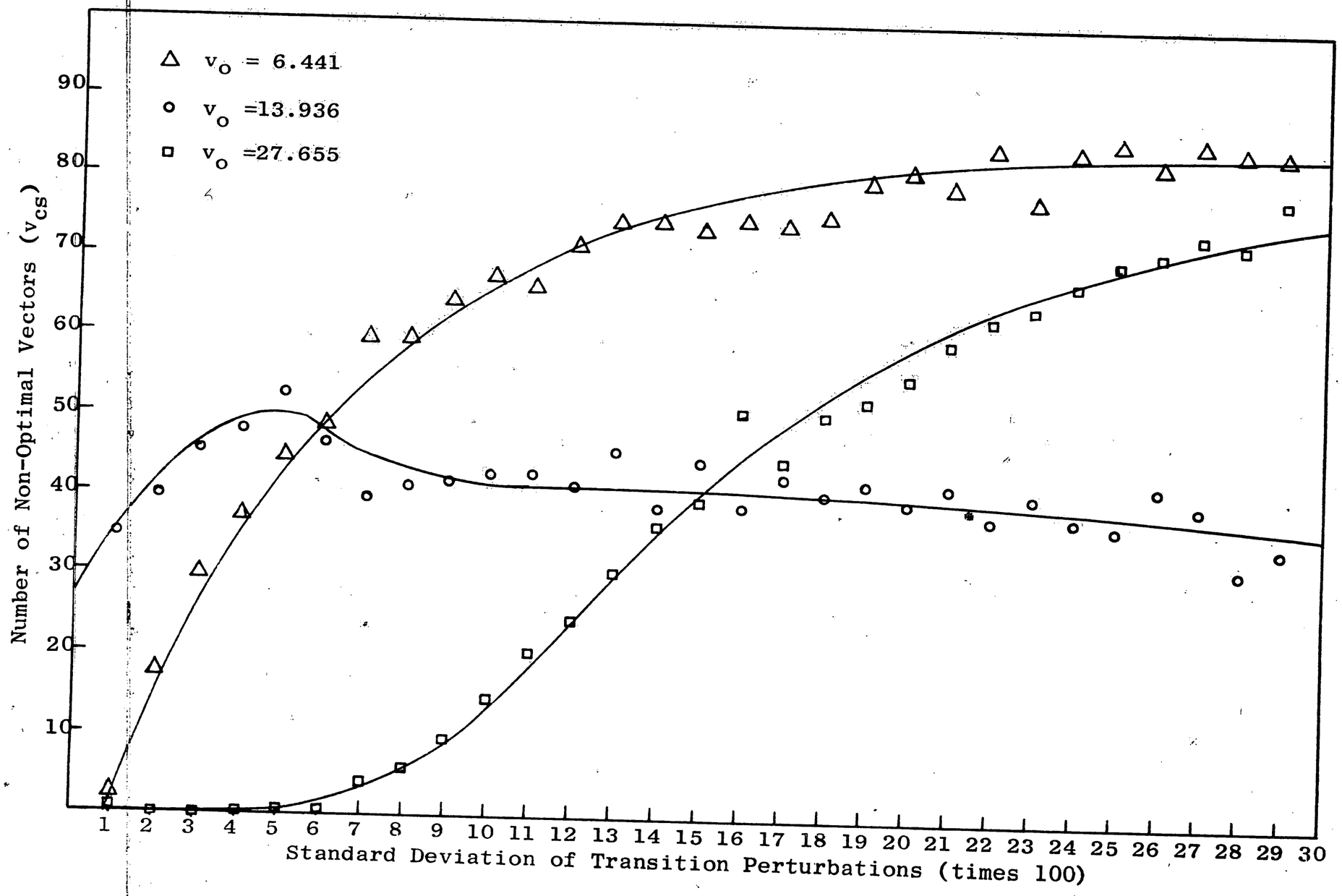
will have one and only one recurrent chain so that by definition the processes are ergodic. This point was mentioned in Chapter II. A further implication of this process will be discussed in the next chapter.

CHAPTER IV

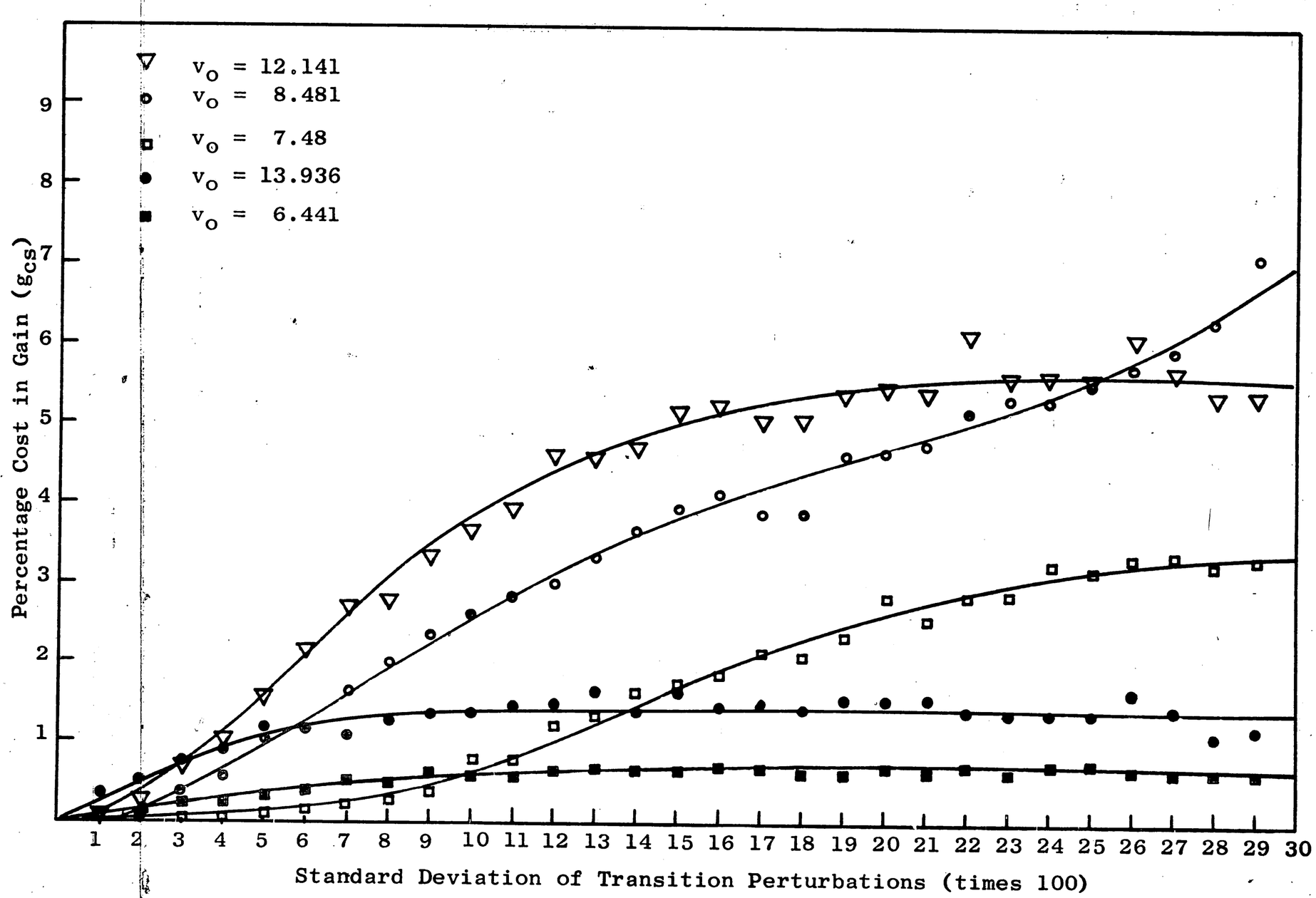
Conclusions

The results of this study indicate that the policy-iteration technique does not exhibit a characteristic sensitivity. Instead we find that the sensitivity is strictly dependent upon the structure of the decision process itself. By this we mean that the sensitivity varies from problem to problem. Furthermore these variations are quite pronounced. Figures 4.1 and 4.2 illustrate this point. Figure 4.1 gives v_{cs} (the number of non-optimal policy vectors selected from a set of one-hundred perturbed systems) as a function of s(the standard deviation of the transition perturbations) for three different Markovian decision problems. These problems are identified by v_o as previously defined. For the problem identified as v_o=27.655 we see that v_{cs} starts out very small and gradually increases to about 70 while for v_o = 13.936, v_{cs} starts out fairly high and remains relatively constant throughout the variations on s. Figure 4.2 illustrates similarly how g_{cs} varies from problem to problem. This point is further illustrated by a variety of problems included in the appendix.

Although we have concluded that the sensitivity of the policy-iteration technique varies from problem to problem it is still possible to predict the sensitivity for a given problem. The method developed for this thesis seems to be a reasonable method of predicting the sensitivity of given system. Figures 4.3, 4.4, 4.5 and 4.6 serve to illustrate this point. Each of these figures gives the

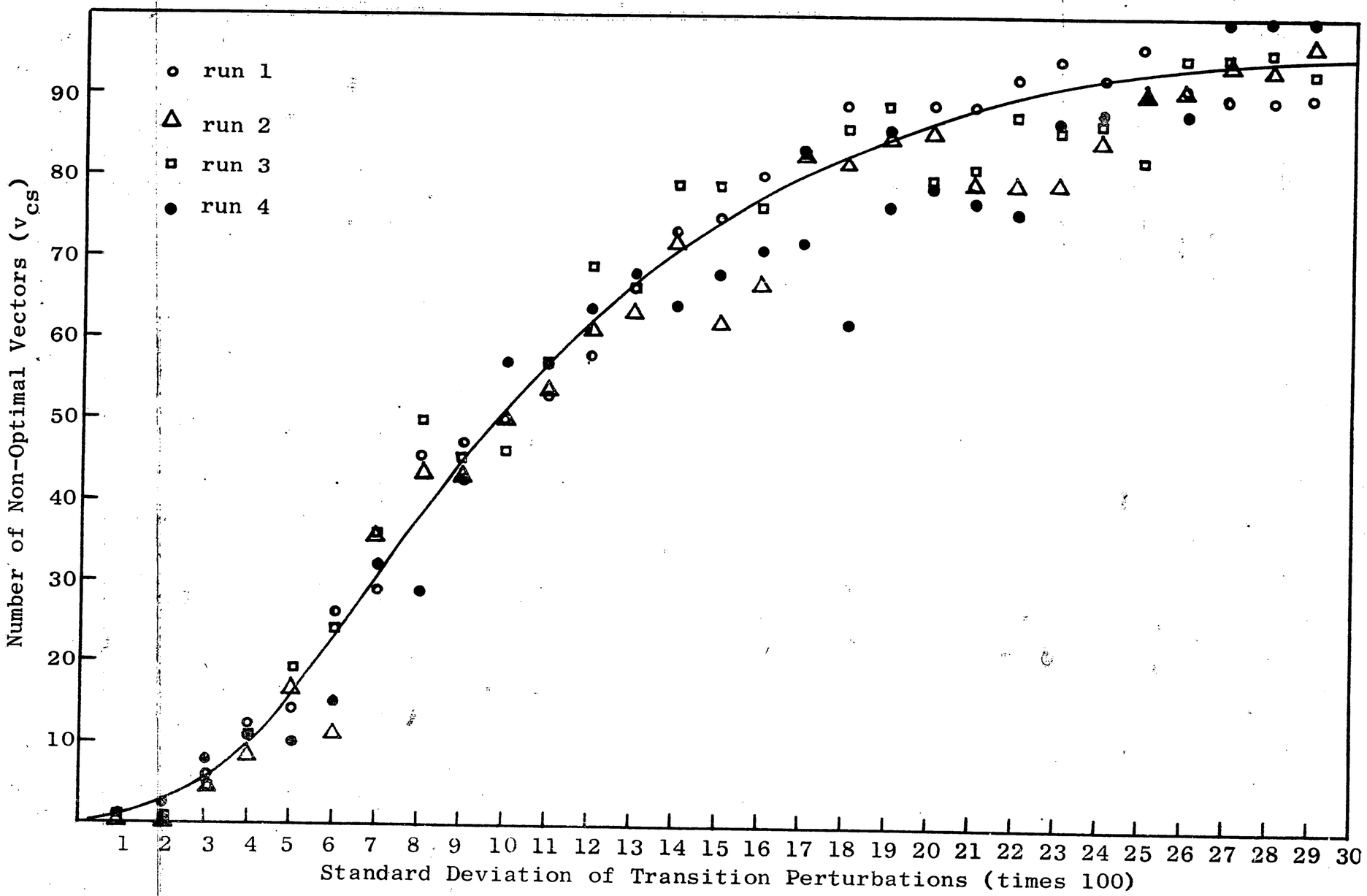


SENSITIVITY CURVES FOR VARIOUS VALUES OF v_o
FIGURE 4.1



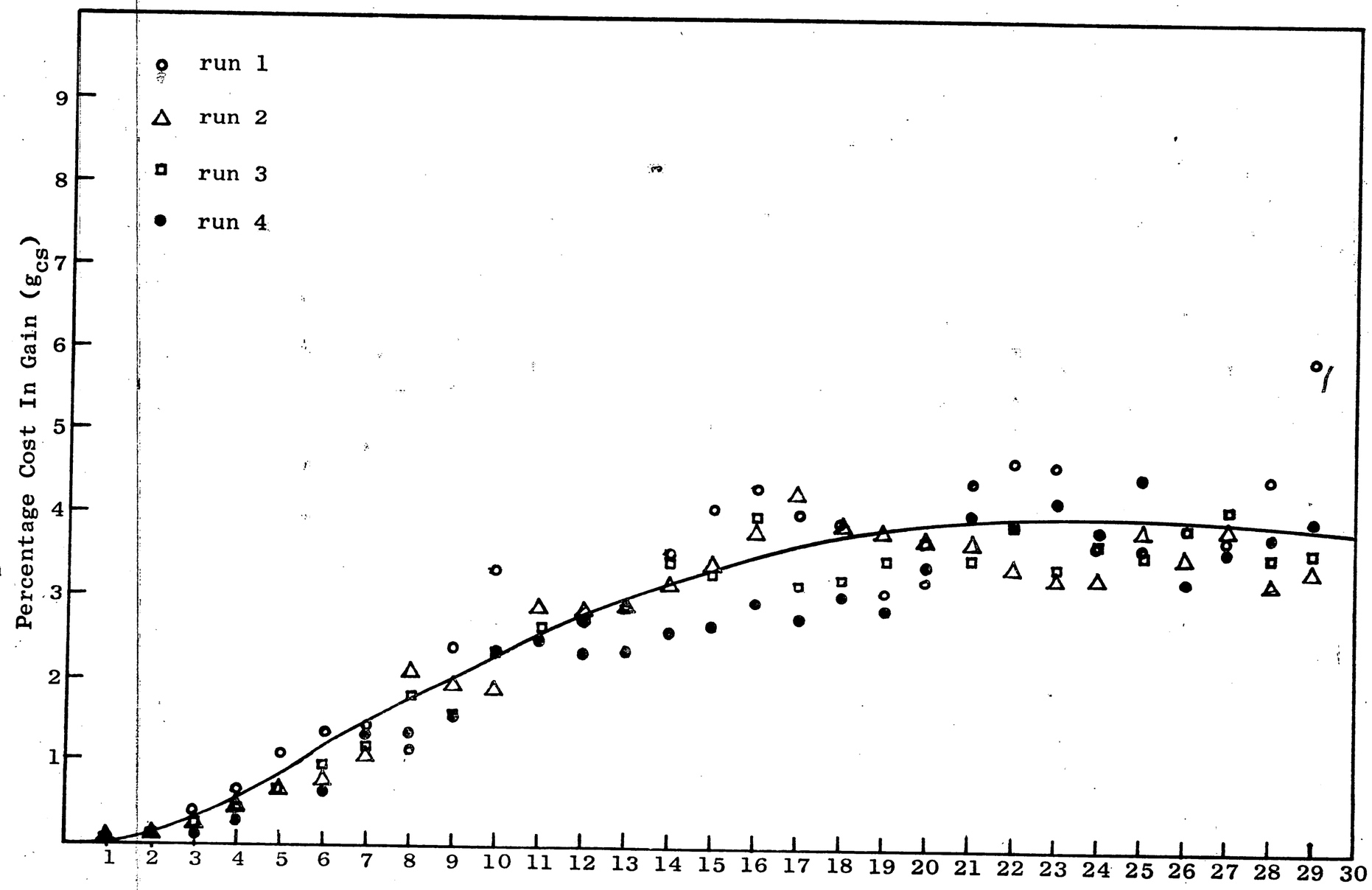
SENSITIVITY CURVES FOR VARIOUS VALUES OF v_o

FIGURE 4.2

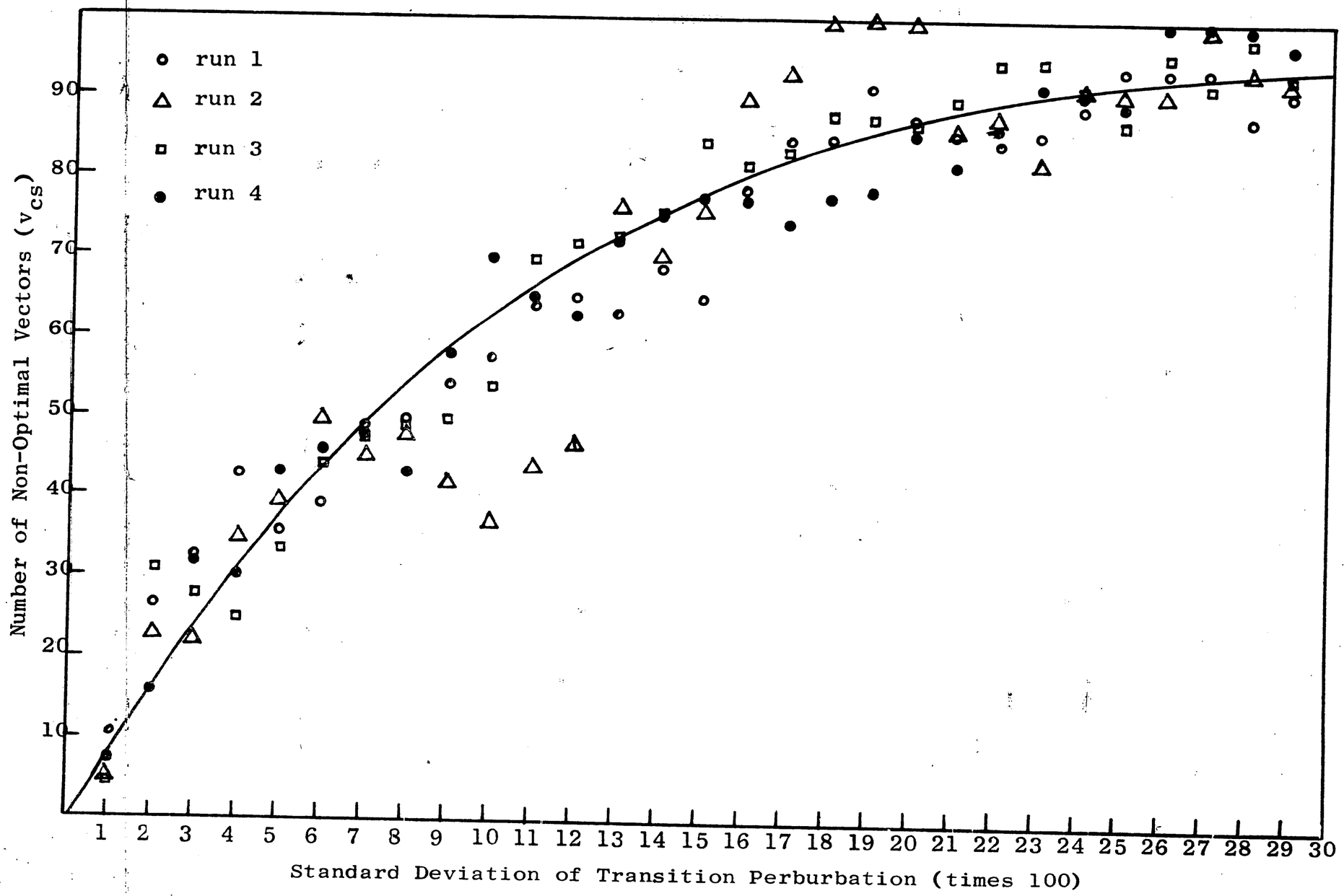


VARIATIONS IN SENSITIVITY FOR DIFFERENT SEQUENCES OF RANDOM NUMBERS

FIGURE 4.3

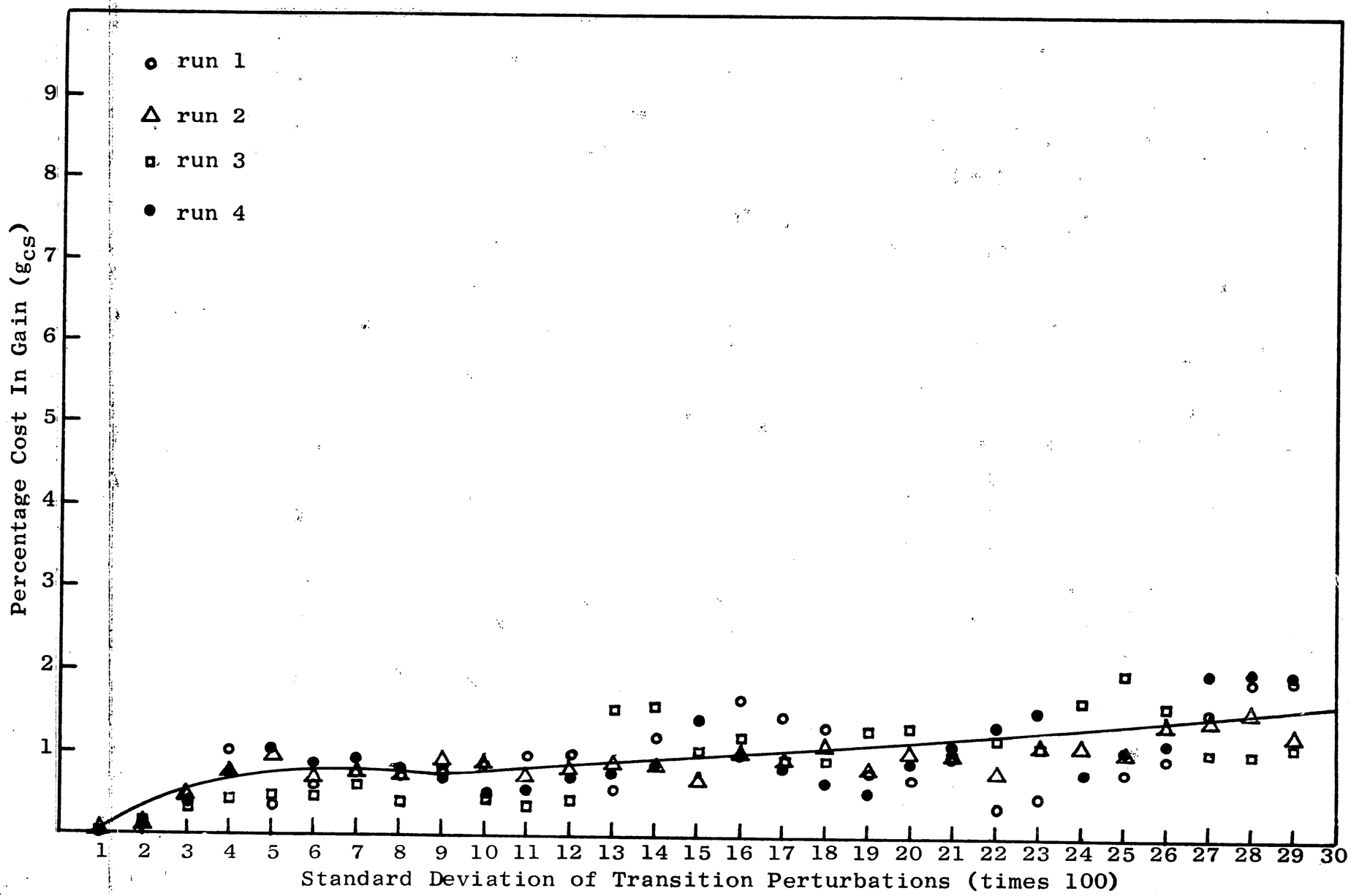


VARIATIONS IN SENSITIVITY FOR DIFFERENT SEQUENCES OF RANDOM NUMBERS
FIGURE 4.4



VARIATIONS IN SENSITIVITY FOR DIFFERENT SEQUENCES OF RANDOM NUMBERS

FIGURE 4.5



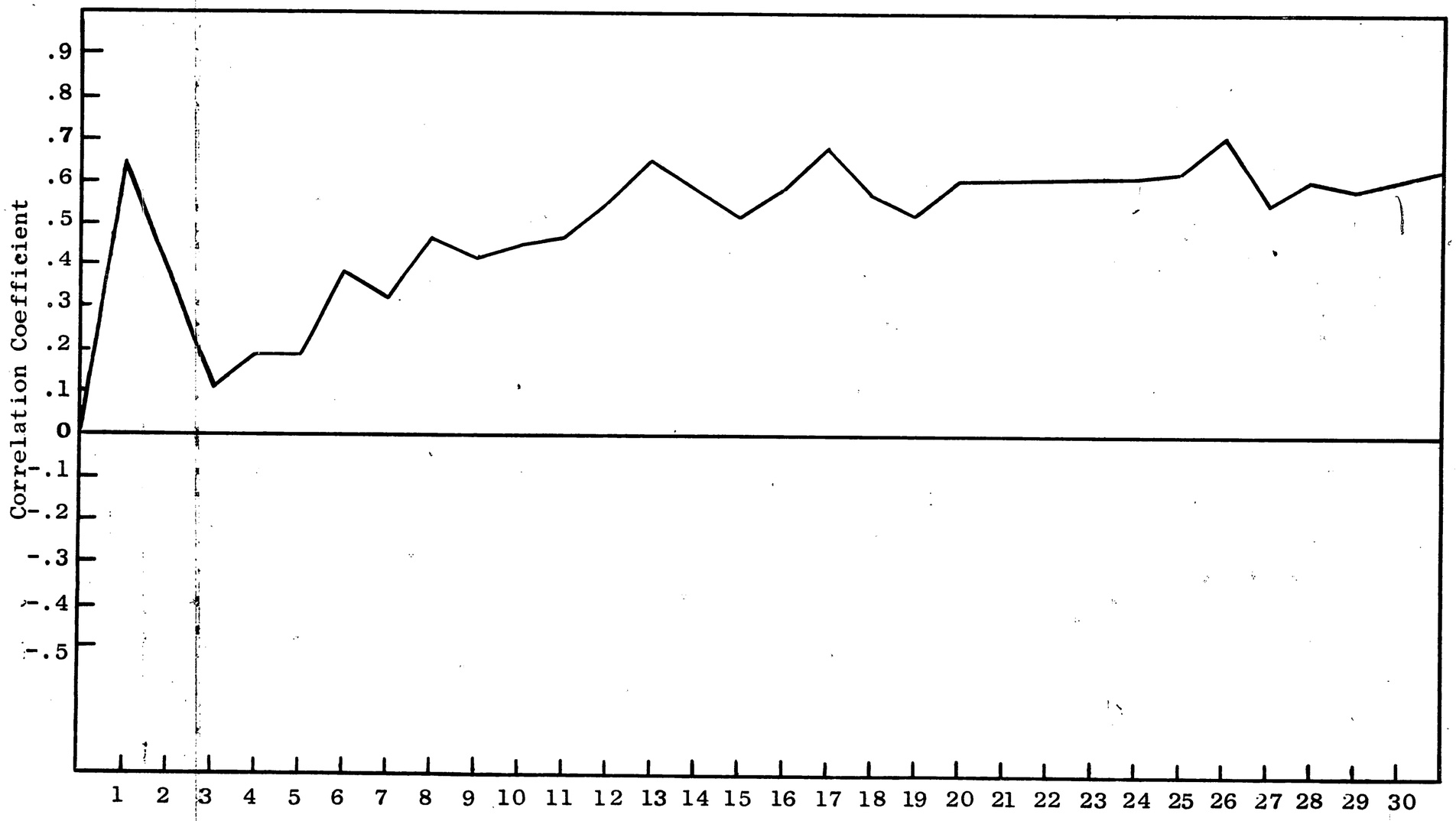
VARIATIONS IN SENSITIVITY FOR DIFFERENT SEQUENCES OF RANDOM NUMBERS

FIGURE 4.6

results of running a given problem with four different sequences of random numbers. This means that the transition perturbations, while having the same control parameters, were actually different. As can be seen from the graphs, essentially the same sensitivity characteristics were obtained in each run. (This point is further illustrated in the appendix. See Figures A.9 thru A.15). Therefore it is possible to estimate the stochastic parameters of a Markovian decision problem and through use of the method presented here to obtain an appreciation for the sensitivity of the system.

The FORTRAN program which implements the method developed for this thesis will remain in the author's files for five years. This is a generalized program and as such, might require minor modification for specific cases. An example of this would be its application to an inventory problem structured as a Markovian decision process. In this case the perturbations would have to be applied in a slightly different fashion due to the repetitiveness of the transition probabilities. Contained within the program is an identifiable module representing a completely generalized implementation of the policy-iteration method. It will accept a variable number of states with a variable number of alternatives in each state. The size of problem it will accommodate is limited only by available core storage.

It is interesting to observe the lack of correlation between v_{cs} and g_{cs}. This is illustrated in Figure 4.7 which shows the correlation coefficient calculated at each value of s. Thirty-nine data points were used at each value of s. From these data we can conclude that a high probability of selecting a non-optimal policy vector does



CORRELATION BETWEEN g_{cs} and v_{cs} FOR VARIOUS VALUES OF s.
(s RANGES FROM .01 TO .31 IN INCREMENTS OF .01)

FIGURE 4.7

not necessarily mean that the error will be costly. These relationships are further illustrated by a sample of results contained in the appendix.

Recommendations for Further Study

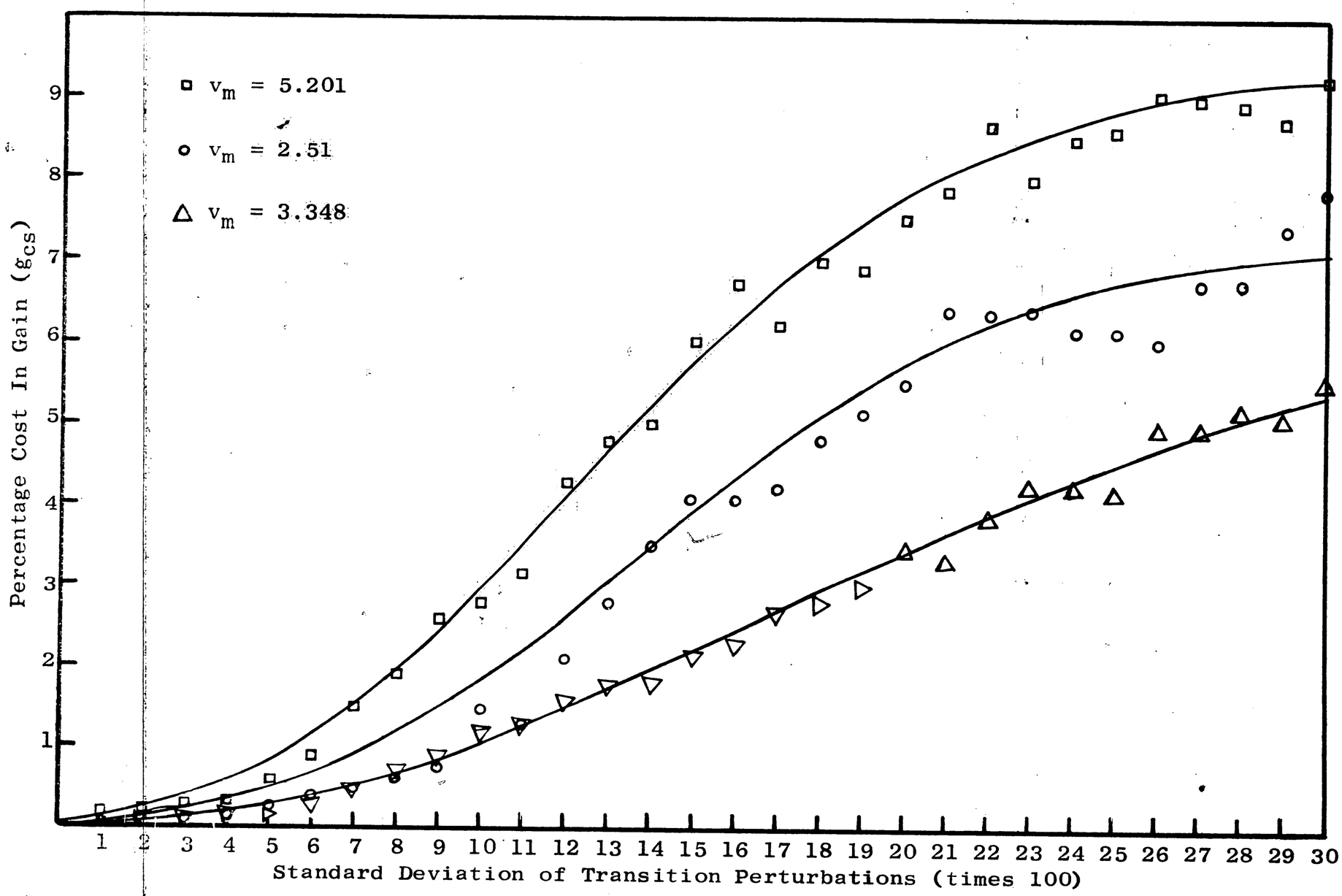
We should briefly mention some areas for further research. One natural extension of this study is suggested. It would be interesting to try to find some characteristic of the decision problems that would correlate to the variations in sensitivity observed. Some of the more intuitively appealing possibilities were investigated during the study. Although the results in this area were negative, it is felt that further efforts would be fruitful. These investigations are discussed in the appendix.

Another area in which further study seems desirable is suggested by the work done by d'Epenoux[1], Manne[8] and Ghellinck[7], et.al. The work of these authors demonstrates the feasibility of applying linear programming techniques to areas previously considered sacred to dynamic programming. Once a sequential decision process is structured as a linear programming model, parametric programming can be used for generalized sensitivity evaluations. Although this may quite possible turn out to be overly cumbersome, it should be investigated.

APPENDIX

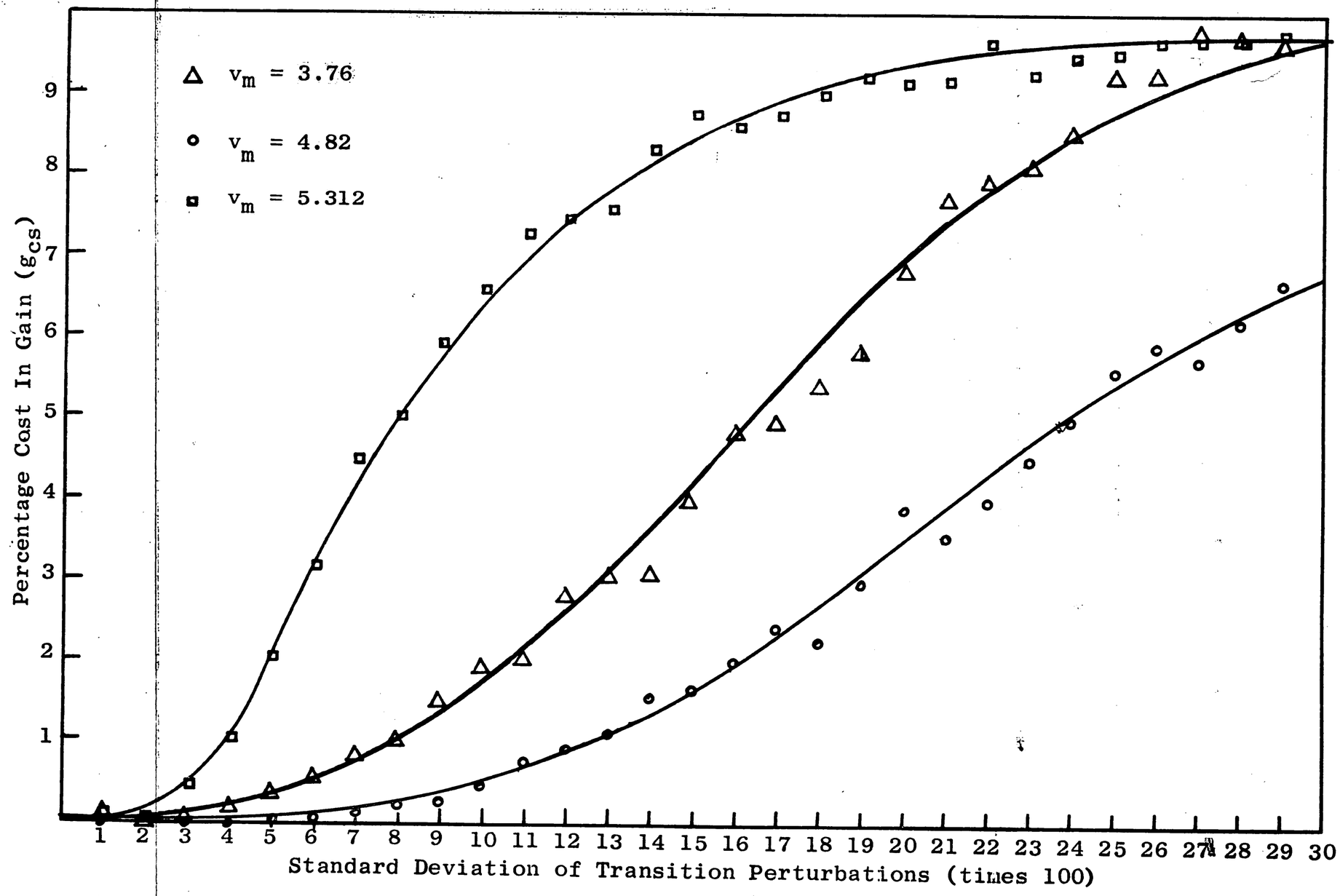
Explanation of Graphs (A.1 - A.8)

The following eight graphs depict the variations in sensitivity observed for a sample of decision problems. The sensitivity is reflected by v_{cs} and g_{cs} as previously defined. In these graphs v_m and v_o serve only as identifiers. These were chosen to identify the various problems as part of the effort to determine if there existed a specific characteristic of the various problem which could be correlated to the observed variations in sensitivity. This effort is discussed in a later section of this appendix.



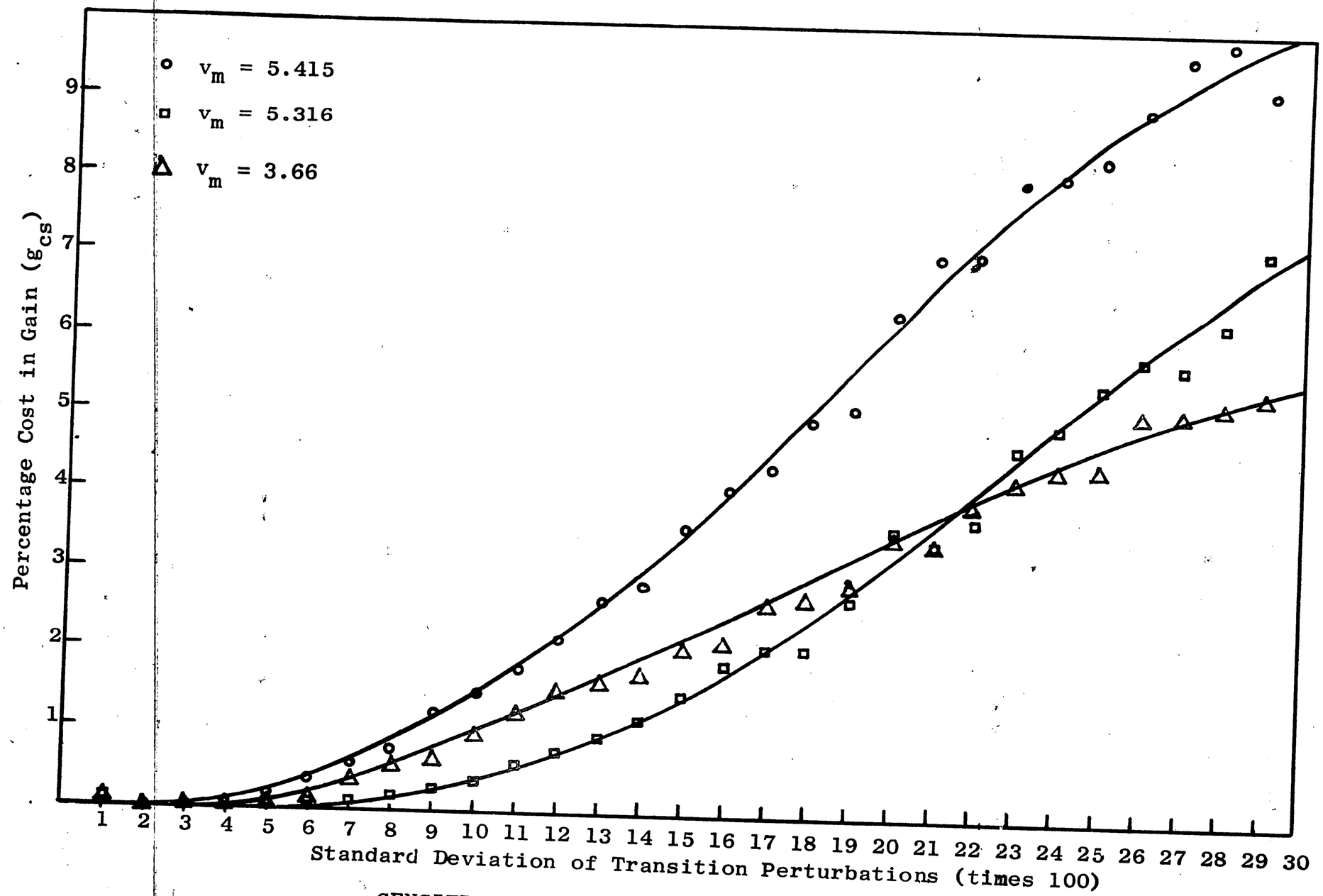
SENSITIVITY CURVES FOR VARIOUS VALUES OF v_m

FIGURE A.1



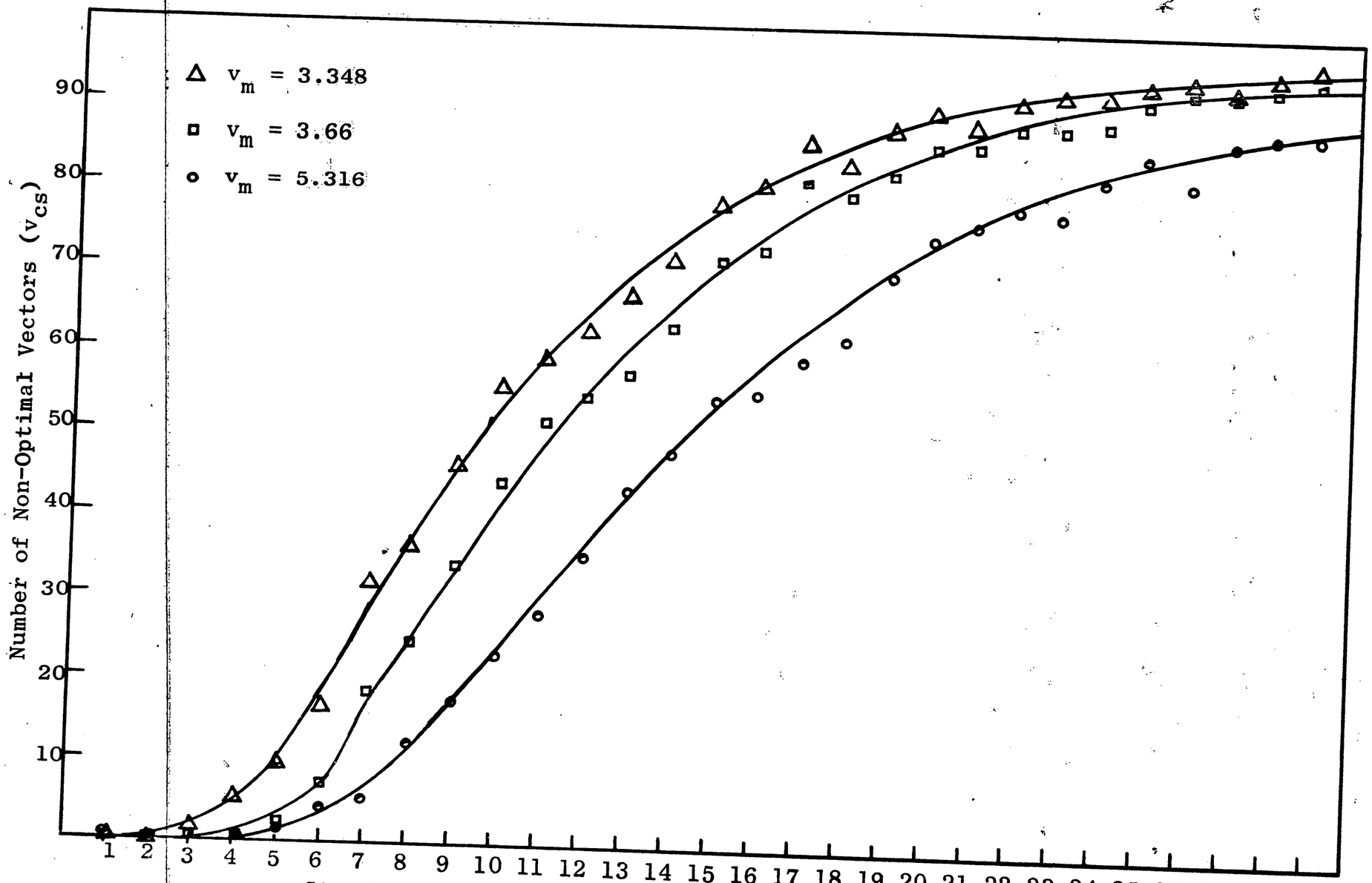
SENSITIVITY CURVES FOR VARIOUS VALUES OF v_m

FIGURE A.2



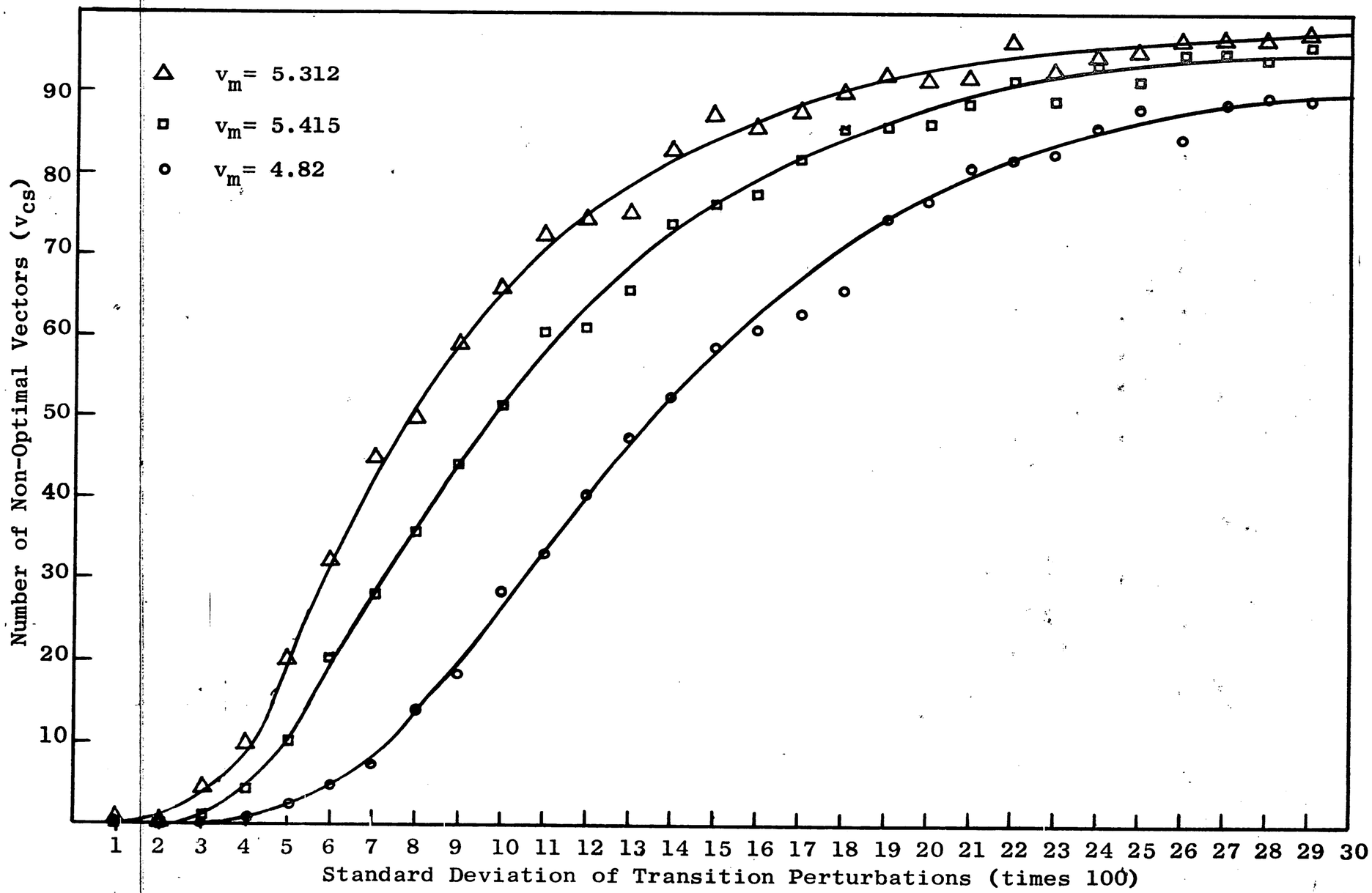
SENSITIVITY CURVES FOR VARIOUS VALUES OF v_m

FIGURE A.3



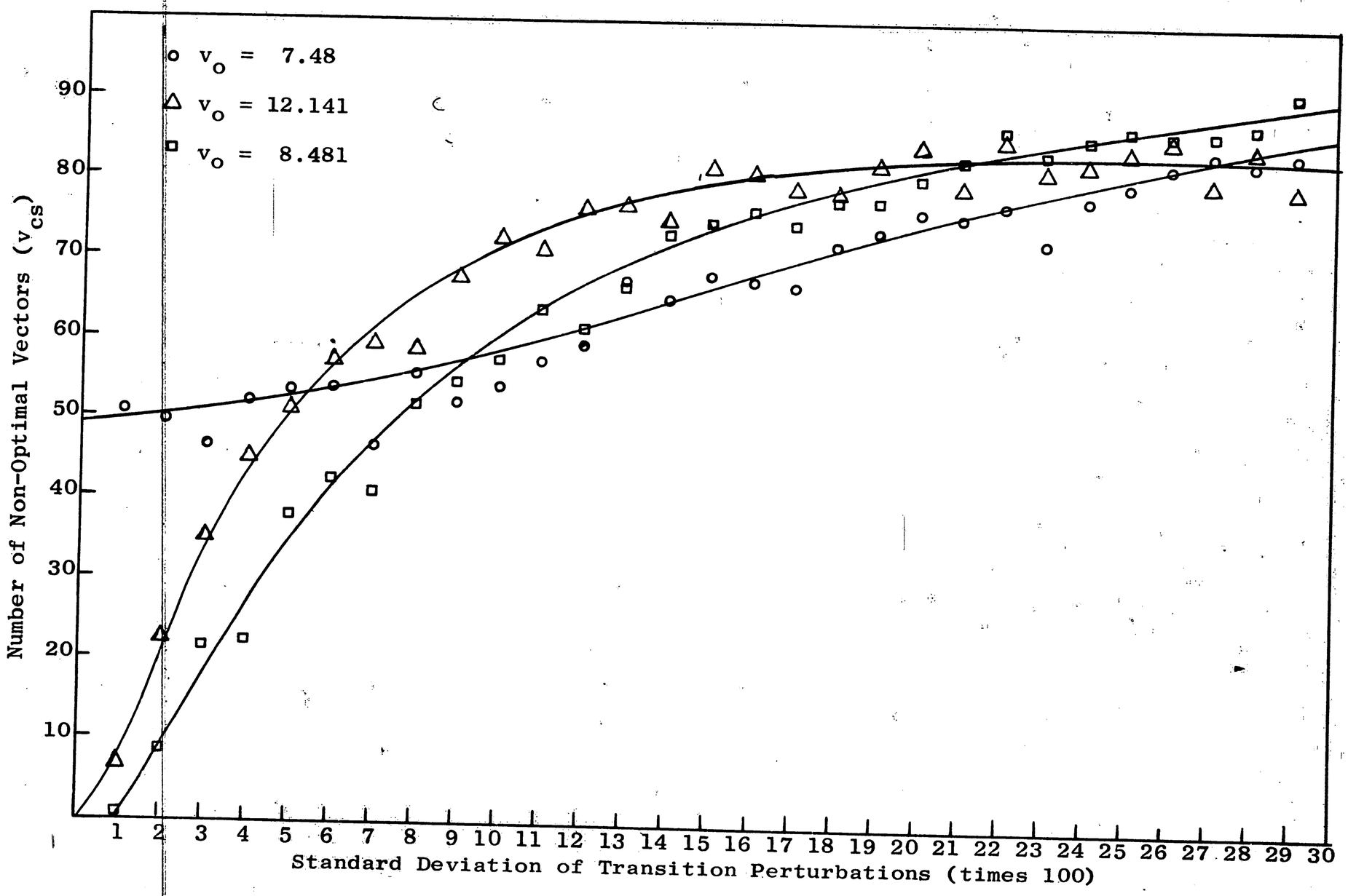
SENSITIVITY CURVES FOR VARIOUS VALUES OF v_m

FIGURE A.4



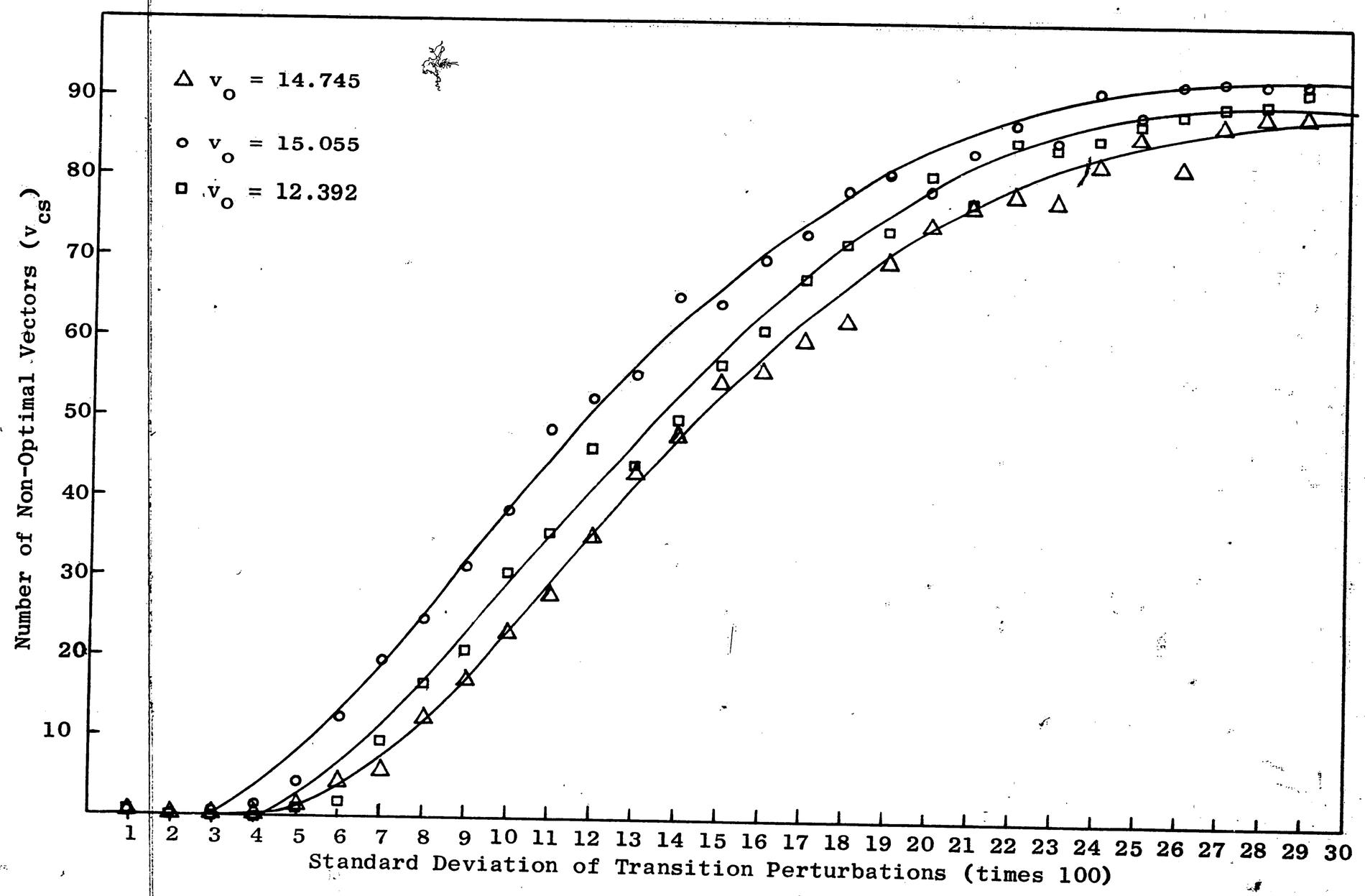
SENSITIVITY CURVES FOR VARIOUS VALUES OF v_m

FIGURE A.5

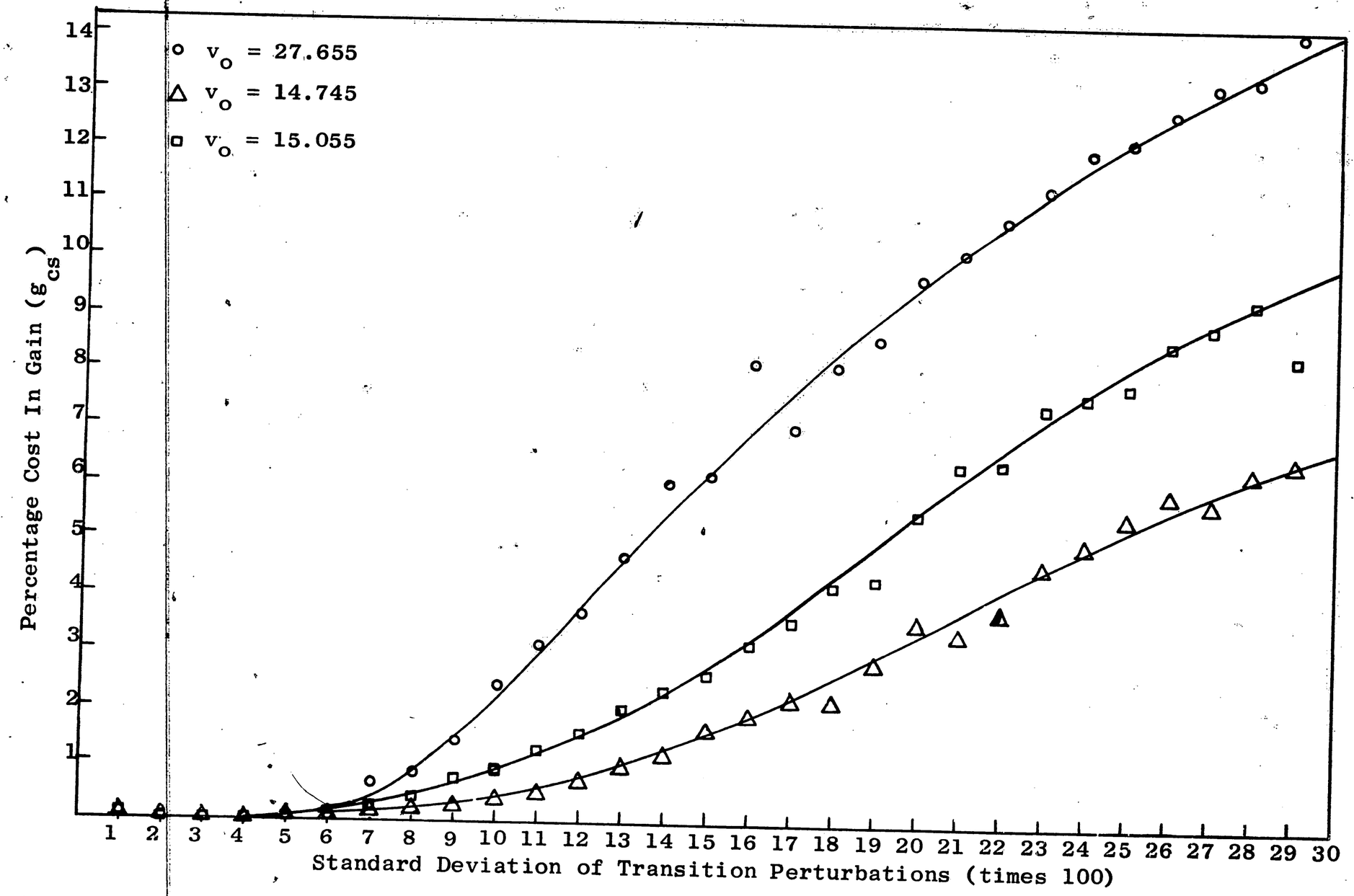


SENSITIVITY CURVES FOR VARIOUS VALUES OF v_o

FIGURE A.6



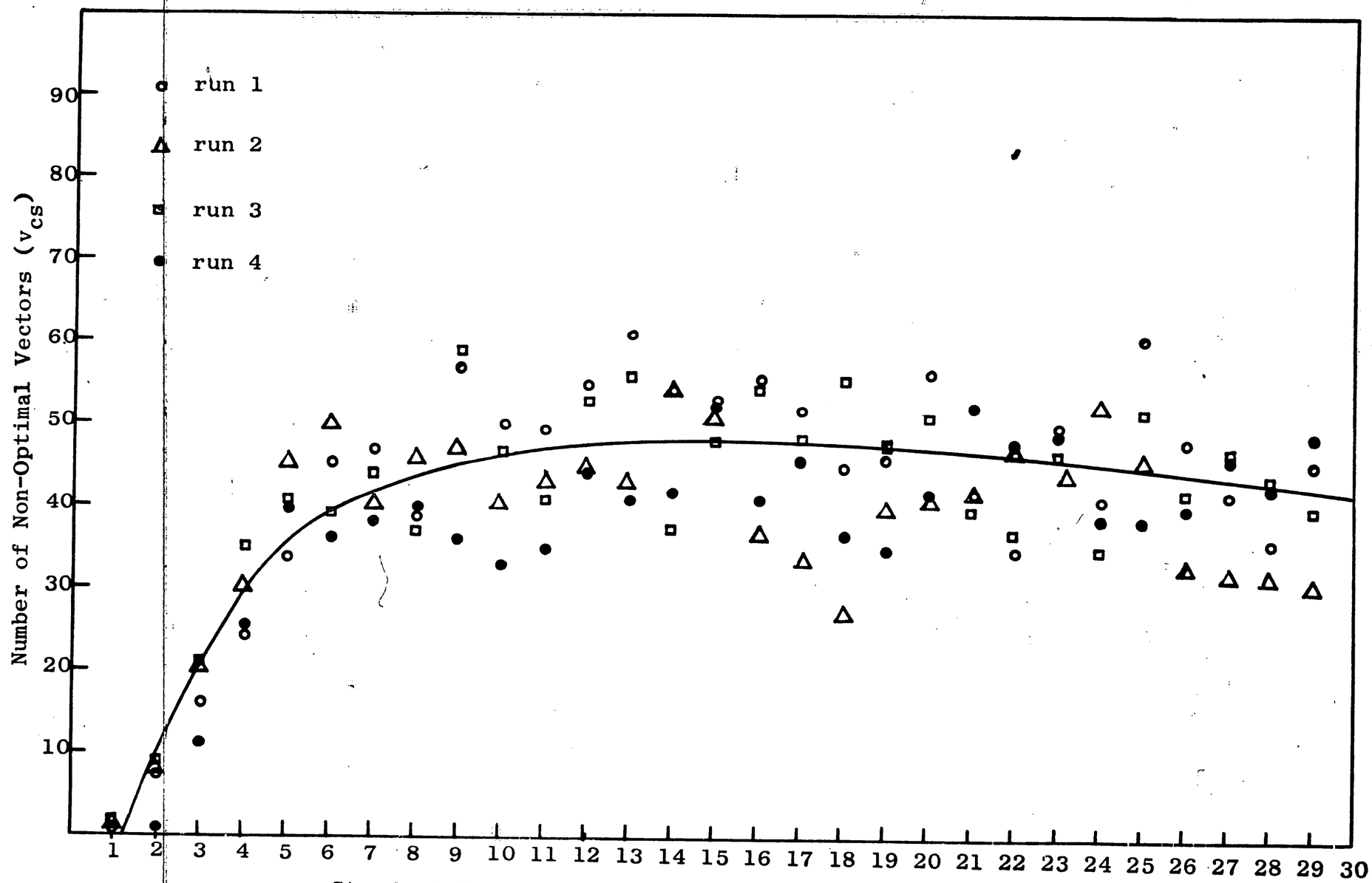
SENSITIVITY CURVES FOR VARIOUS VALUES OF v_o
FIGURE A.7



SENSITIVITY CURVES FOR VARIOUS VALUES OF v_o
FIGURE A.8

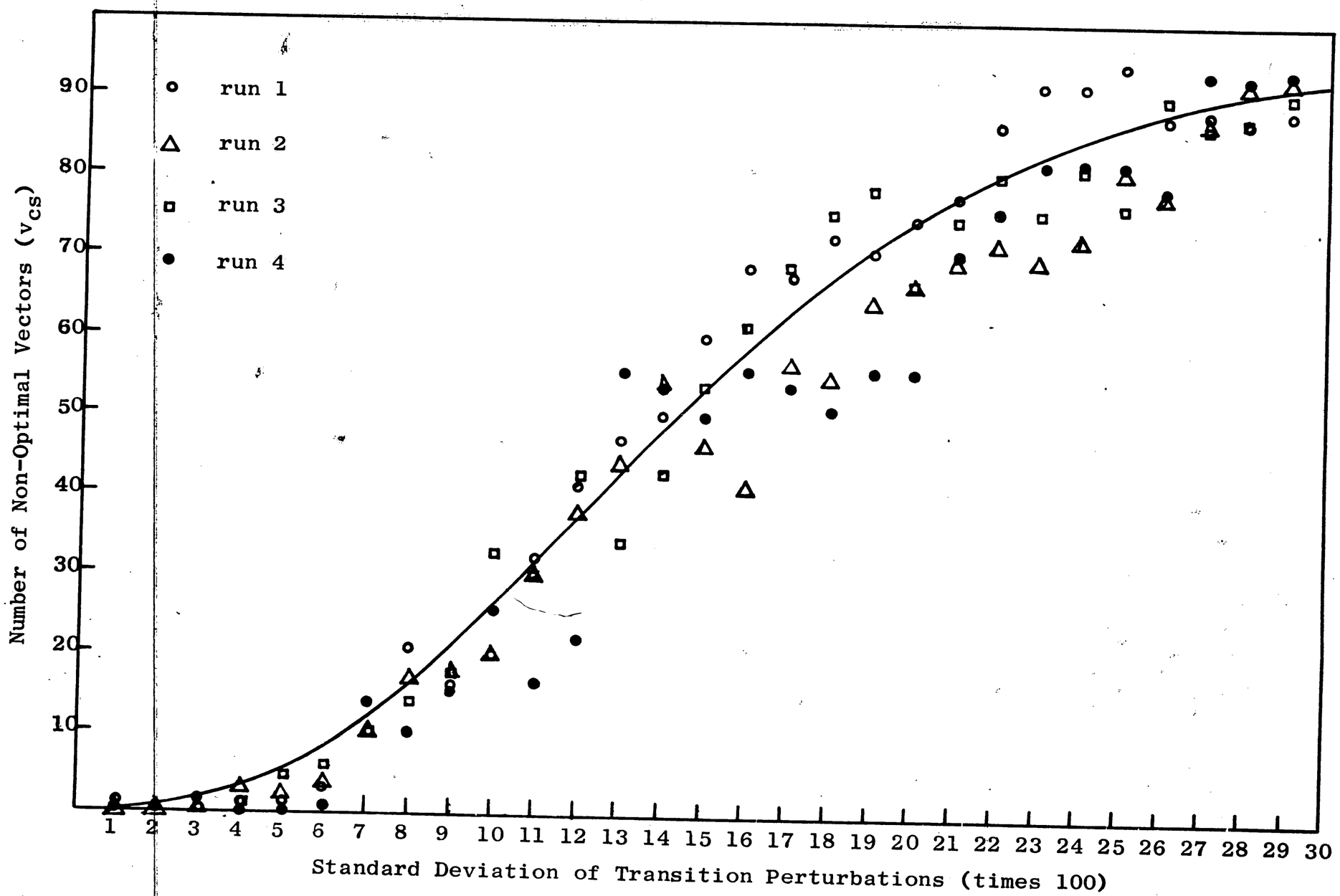
Explanation of Graphs (A.9 - A.15)

Graphs A.9 through A.15 illustrate the repeatability of the sensitivity characteristics of a given Markovian decision problem as measured by the method developed for this thesis. Each graph gives the results of four runs. Each run utilized a unique sequence of random numbers to generate the transition perturbations.



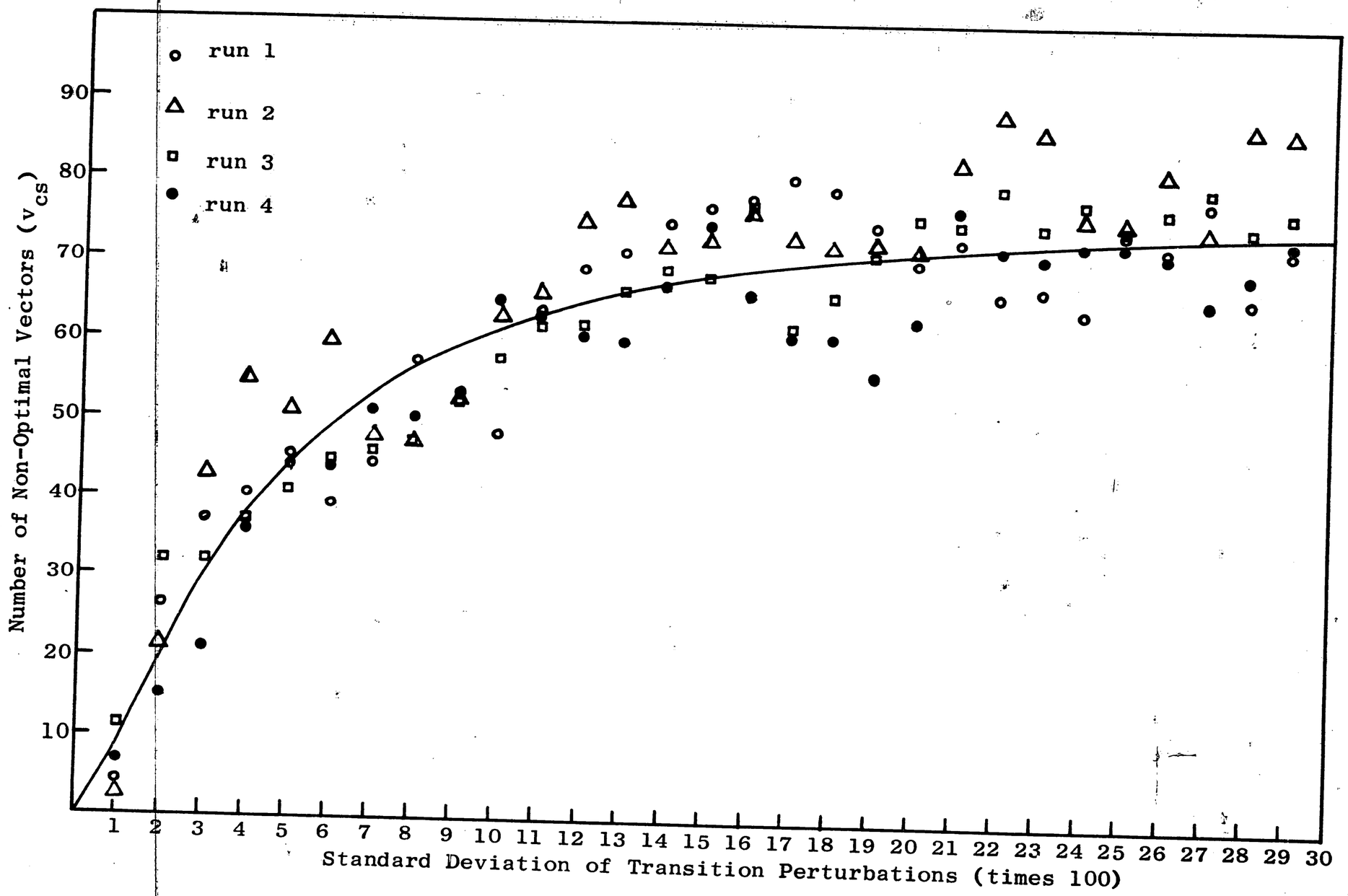
VARIATIONS IN SENSITIVITY FOR DIFFERENT SEQUENCES OF RANDOM NUMBERS

FIGURE A.9



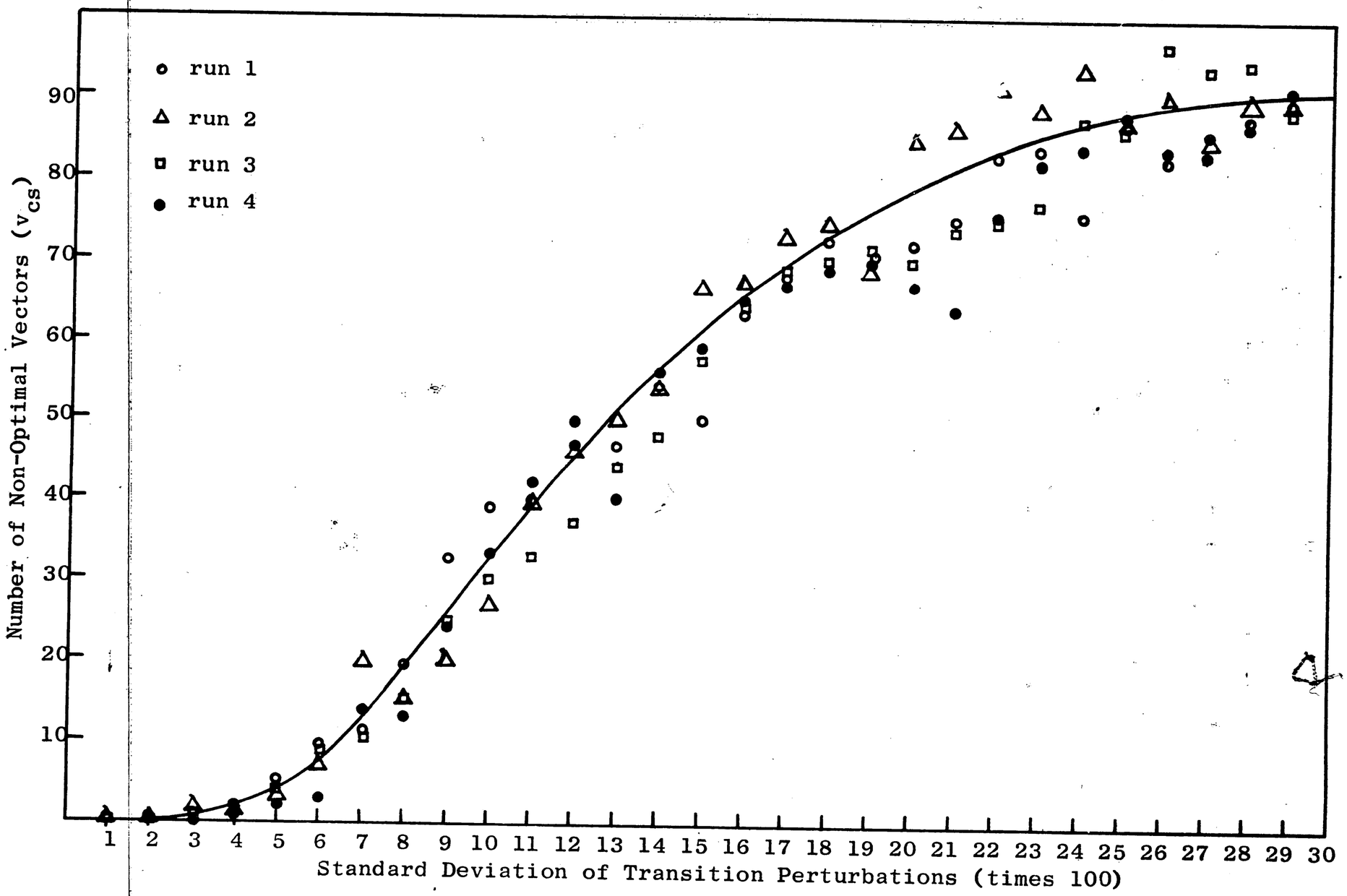
VARIATIONS IN SENSITIVITY FOR DIFFERENT SEQUENCES OF RANDOM NUMBERS

FIGURE A.10



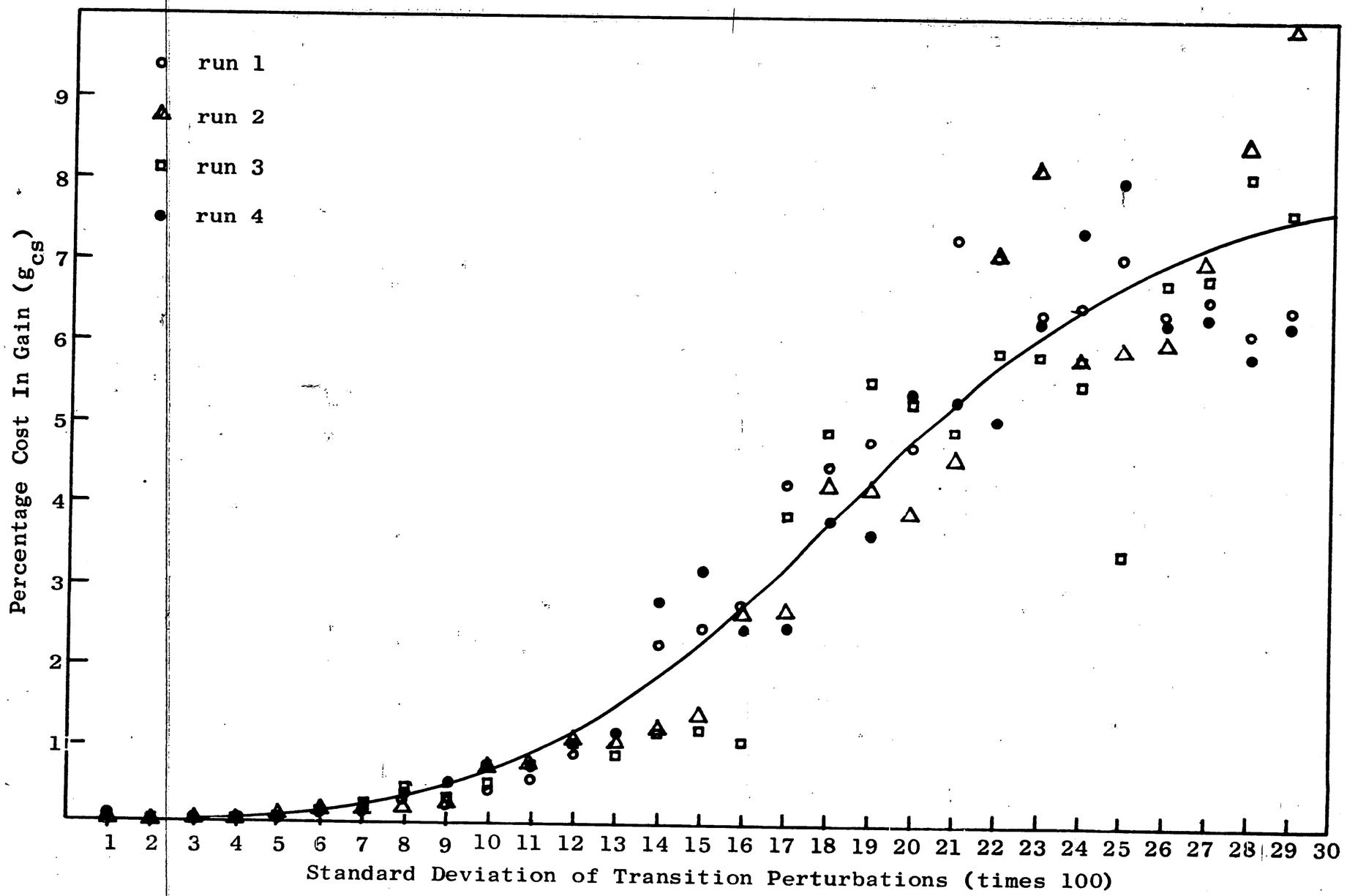
VARIATIONS IN SENSITIVITY FOR DIFFERENT SEQUENCE OF RANDOM NUMBERS

FIGURE A.11



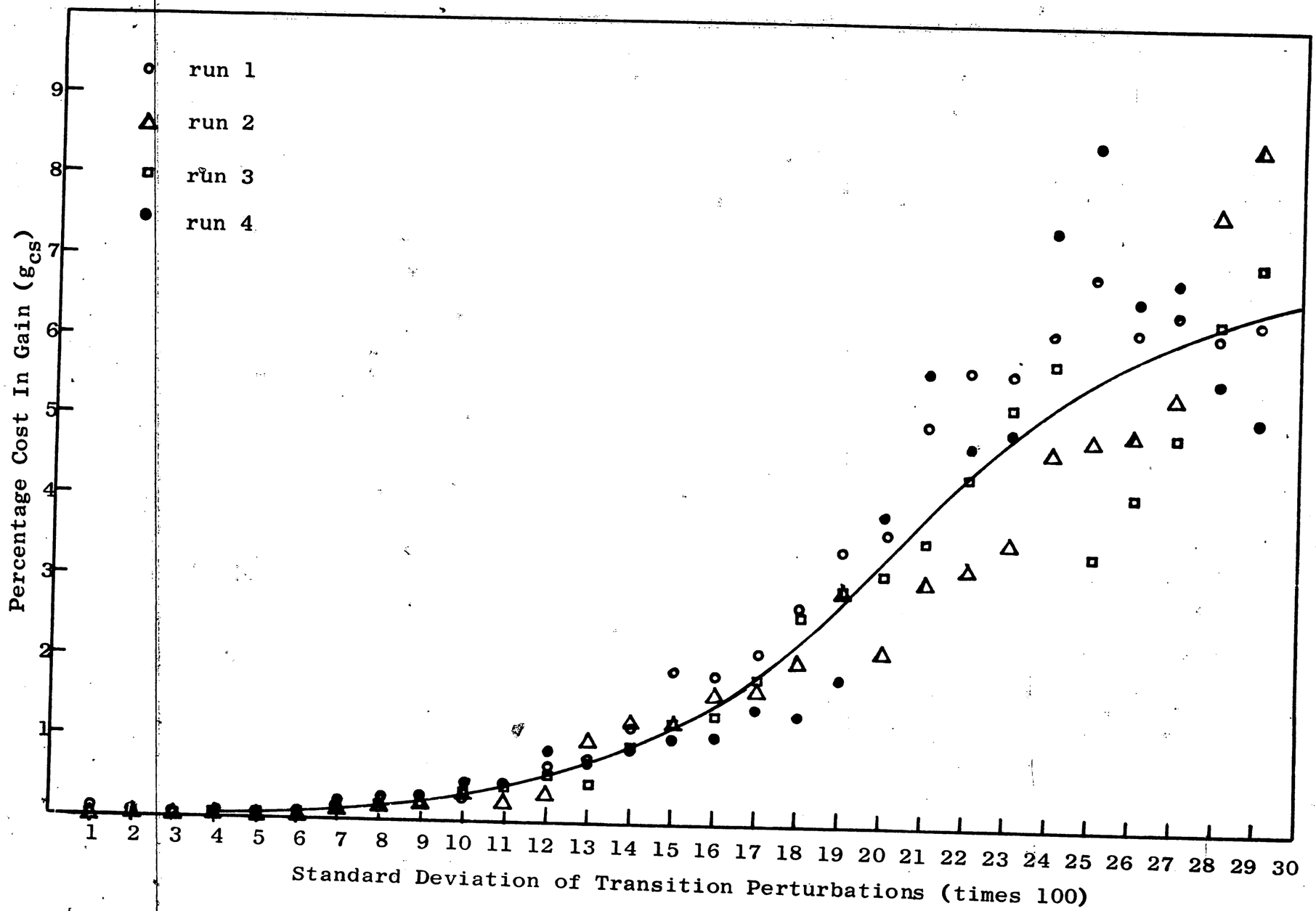
VARIATIONS IN SENSITIVITY FOR DIFFERENT SEQUENCES OF RANDOM NUMBERS

FIGURE A.12



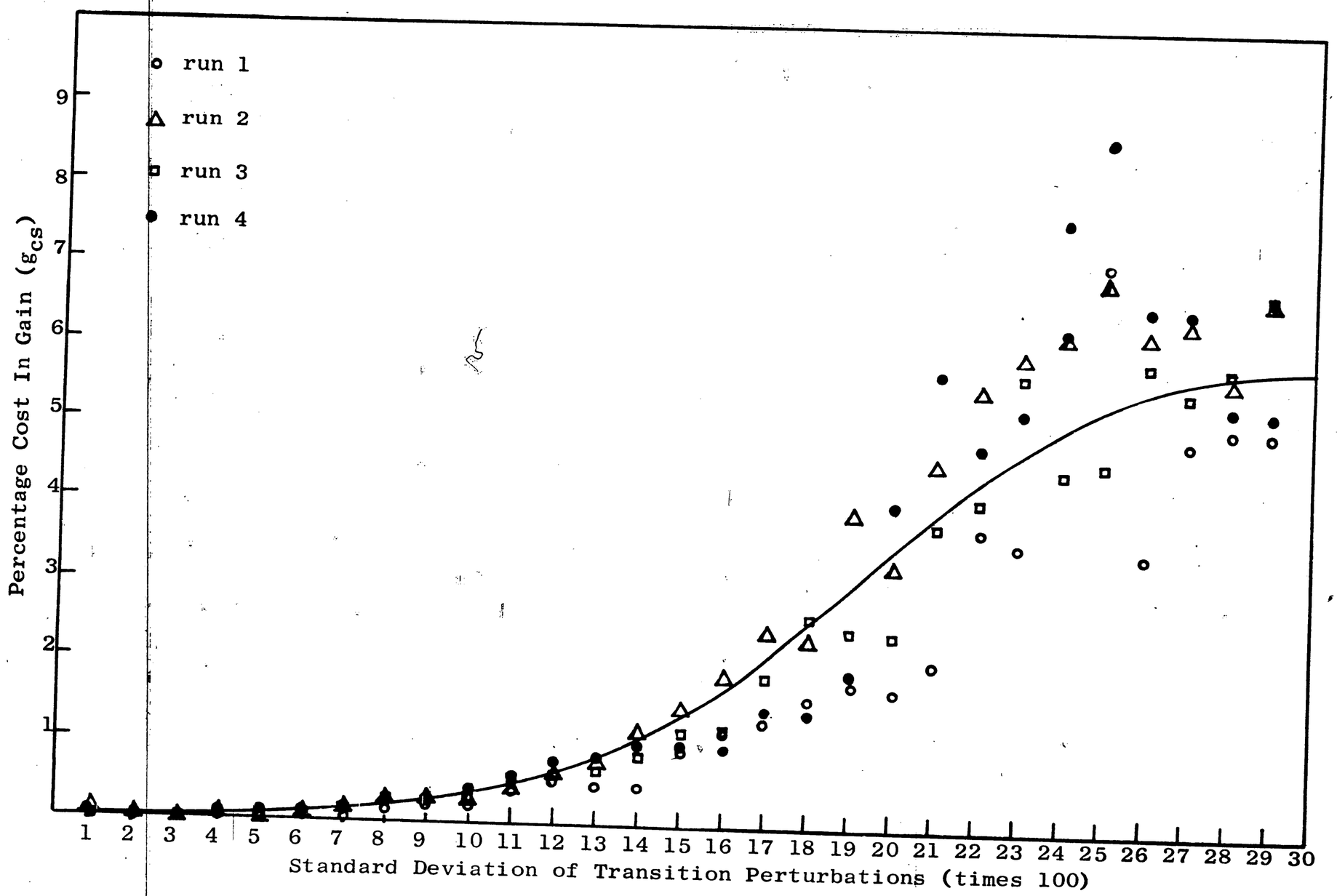
VARIATIONS IN SENSITIVITY FOR DIFFERENT SEQUENCES OF RANDOM NUMBERS

FIGURE A.13



VARIATIONS IN SENSITIVITY FOR DIFFERENT SEQUENCES OF RANDOM NUMBERS

FIGURE A.14



VARIATIONS IN SENSITIVITY FOR DIFFERENT SEQUENCES OF RANDOM NUMBERS

FIGURE A.15

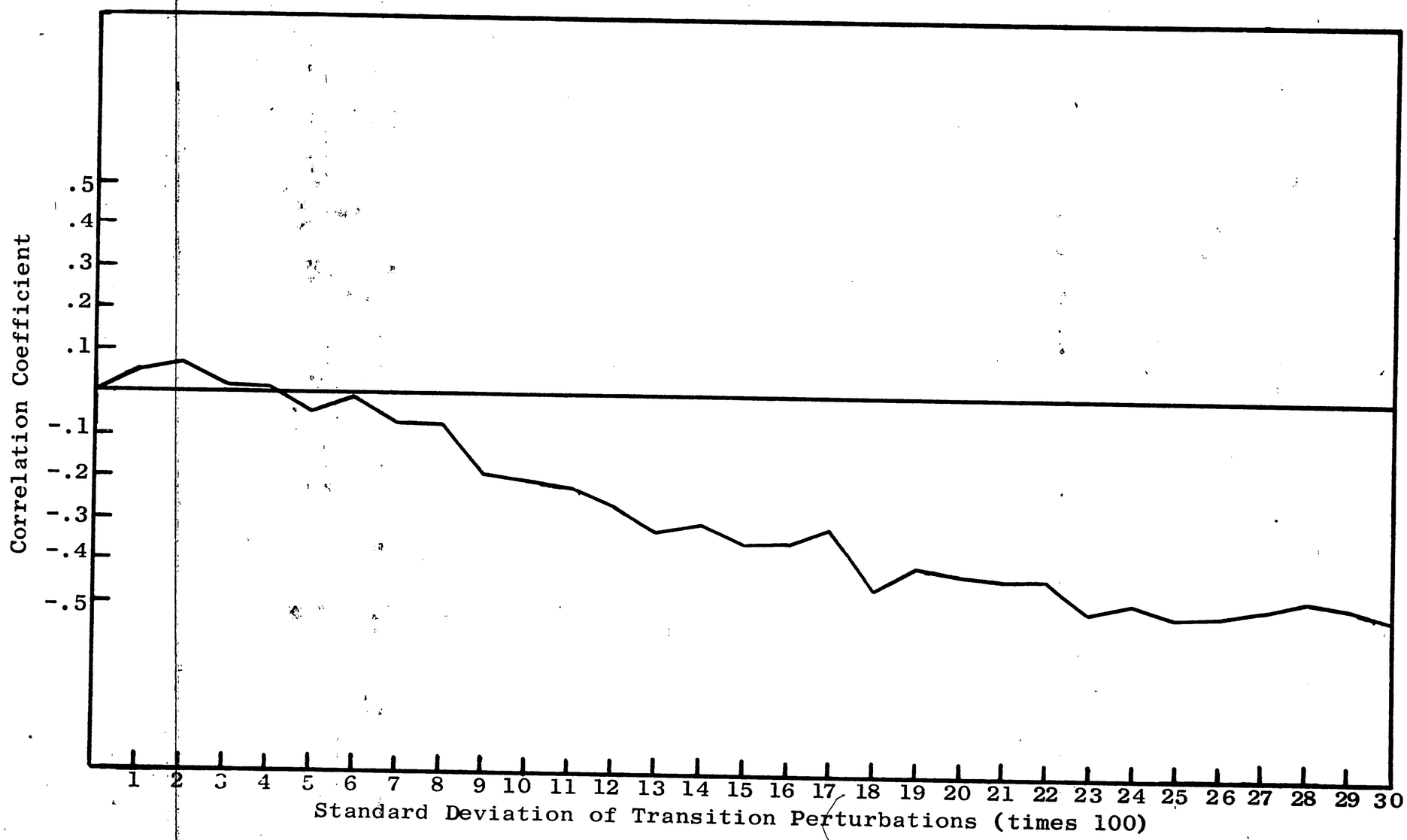
Explanation of Graphs (A.16 - A.23)

Graphs A.16 through A.21 illustrate the lack of correlation between three selected parameters (v_m, v_o and δ) and measured sensitivity for values of s from s = .01 to s = .30. The strongest correlation was between v_{cs} and δ which at best was only slightly below -.7. Although a random sample of problems does not indicate a strong correlation between these two quantities, it was observed that for very small values of δ (i.e. δ = .005) v_{cs} was quite high for small values of s. It was also noted that for small values of s, g_{cs} and v_{cs} were totally un-correlated. Figure A.22 shows the sensitivity curves for a problem having a δ = .004. Here we see the high v_{cs} for small perturbations while the percentage cost in gain remains near zero for relatively large transition perturbations.

Figure A.23 shows the sensitivity curves obtained for a problem having a δ = 4.253 which is near the other extreme. (Most problems from a random sample exhibited a δ considerably smaller than this.) The behavior of g_{cs} is similar to that for the previous problem while v_{cs} behaves quite differently. In all cases where extreme values of δ were observed the behavior of v_{cs} and g_{cs} was essentially the same as that described here.

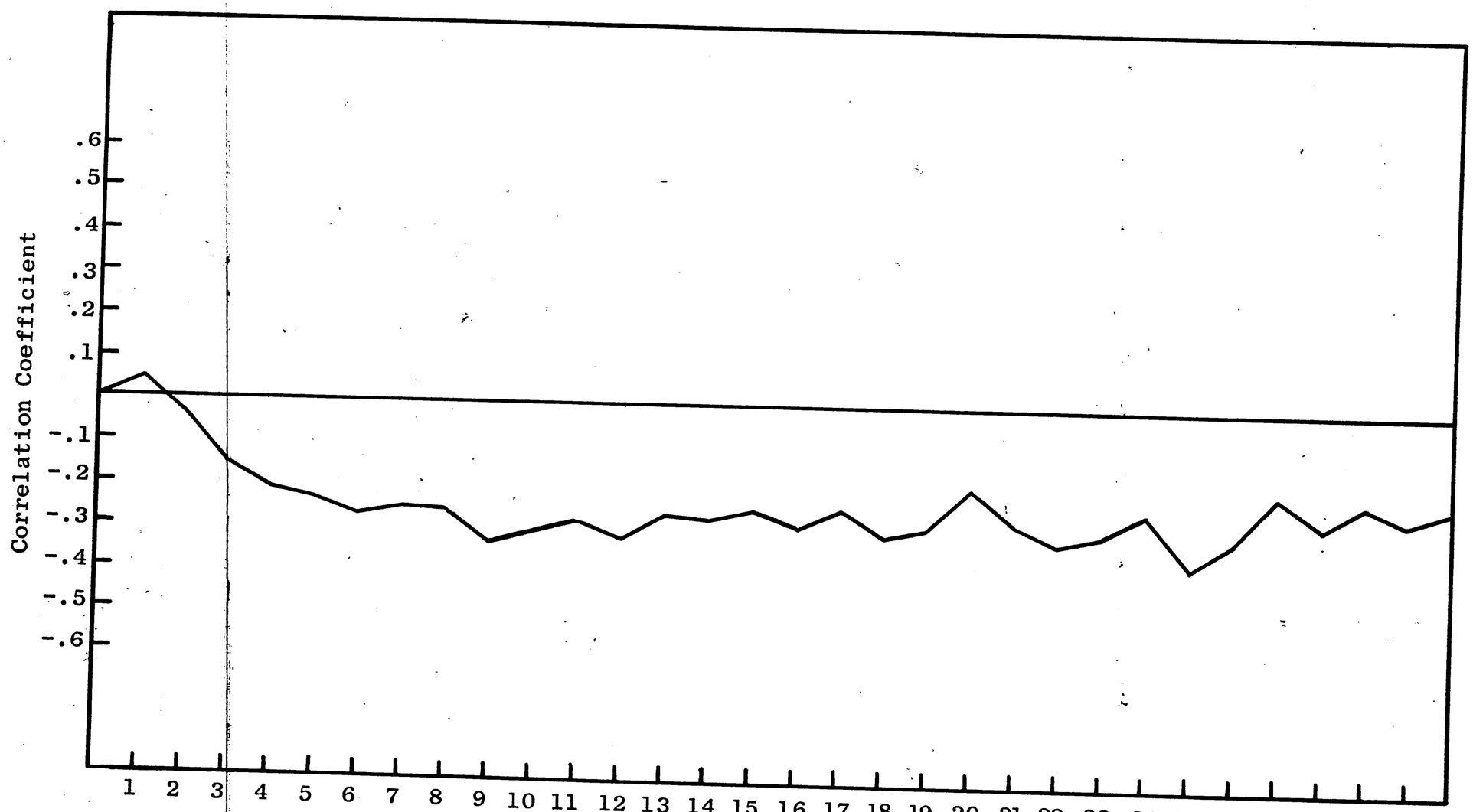
In summary this would seem to indicate that g_{cs} is a reliable and meaningful measure of the system sensitivity for a given decision process and does not follow the eratic behavior of v_{cs} for extreme values of δ. We repeat that this general area (i.e. attempting to correlate a specific characteristic of the decision process

with observed variations in sensitivity) is one to which further attention might be directed.



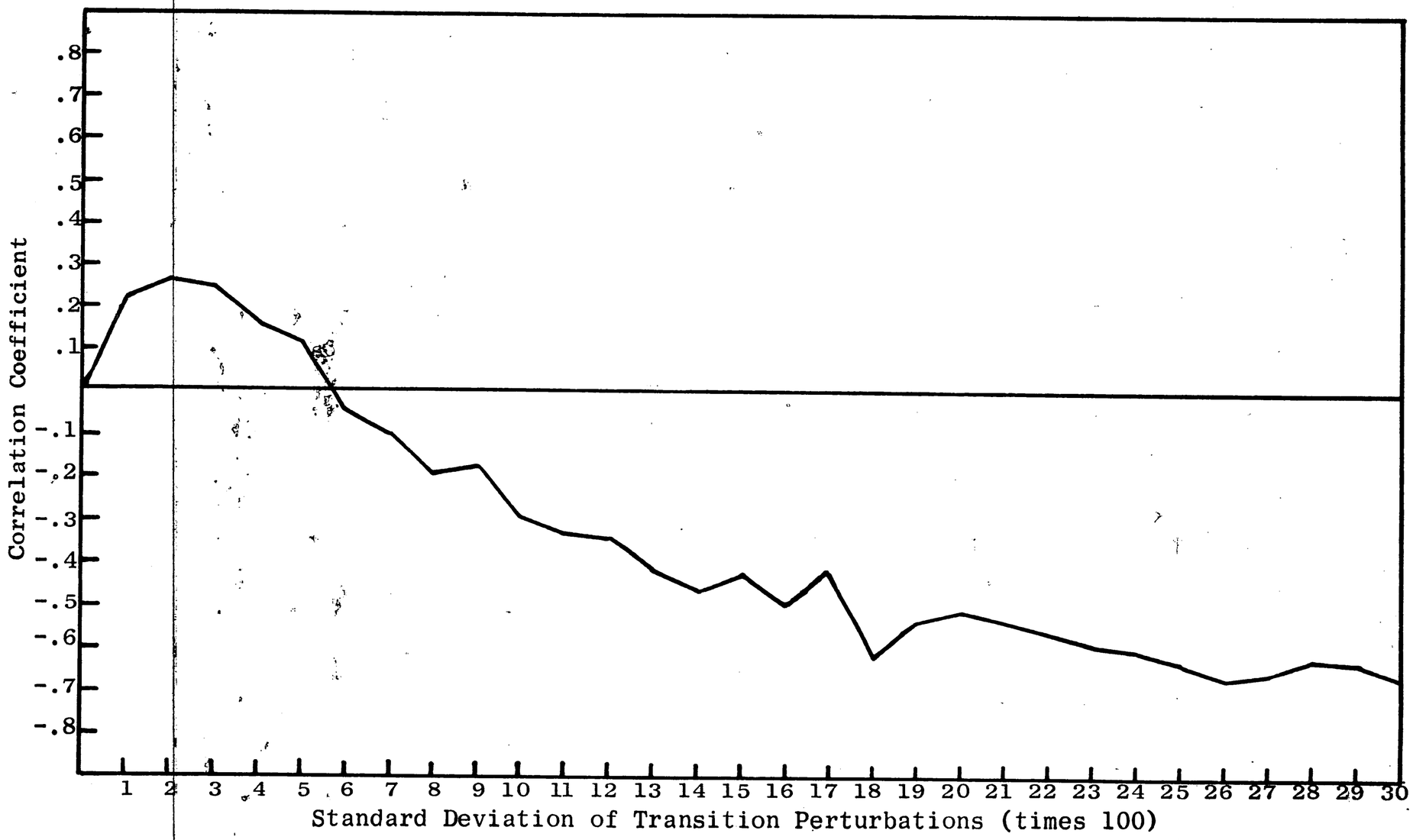
CORRELATION BETWEEN v_{cs} AND v_o FOR VARIOUS VALUES OF s
(s RANGES FROM .01 TO .31 IN INCREMENTS OF .01)

FIGURE A.16



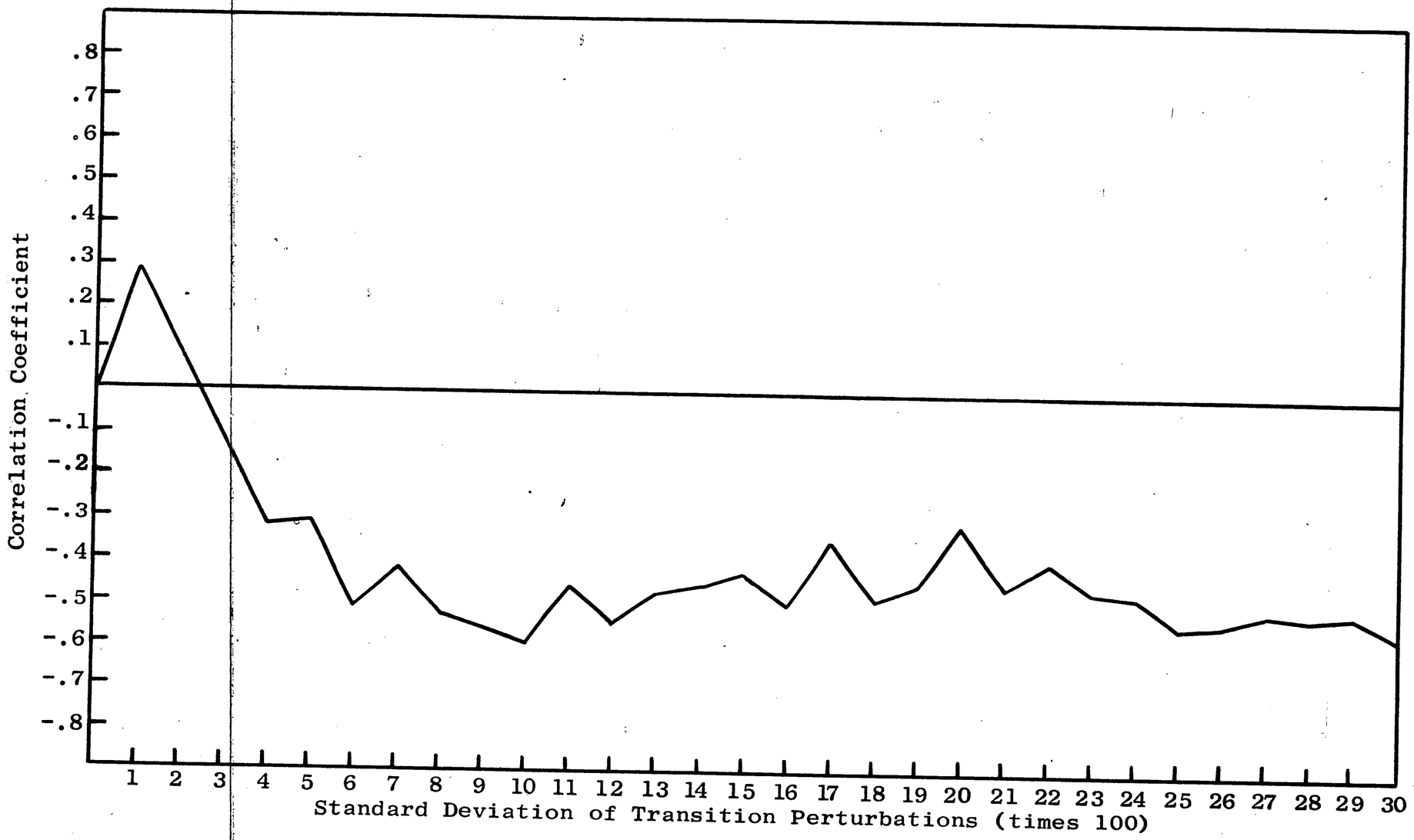
CORRELATION BETWEEN g_{cs} AND v_o FOR VARIOUS VALUES OF s
(s RANGES FROM .01 TO .31 IN INCREMENTS OF .01)

FIGURE A.17



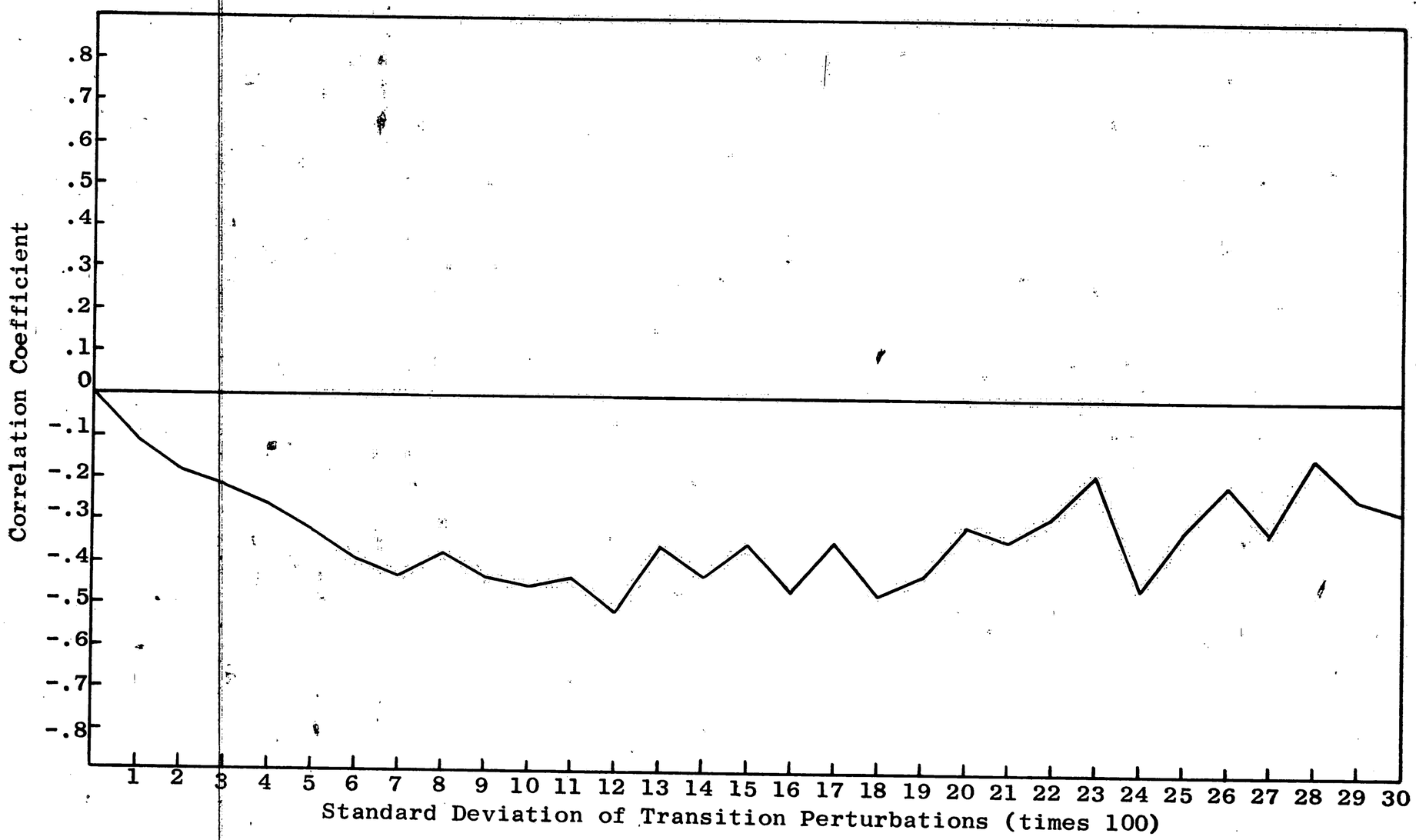
CORRELATION BETWEEN v_{cs} AND v_m FOR VARIOUS VALUES OF s
(s RANGES FROM .01 TO .30 IN INCREMENTS OF .01)

FIGURE A.18



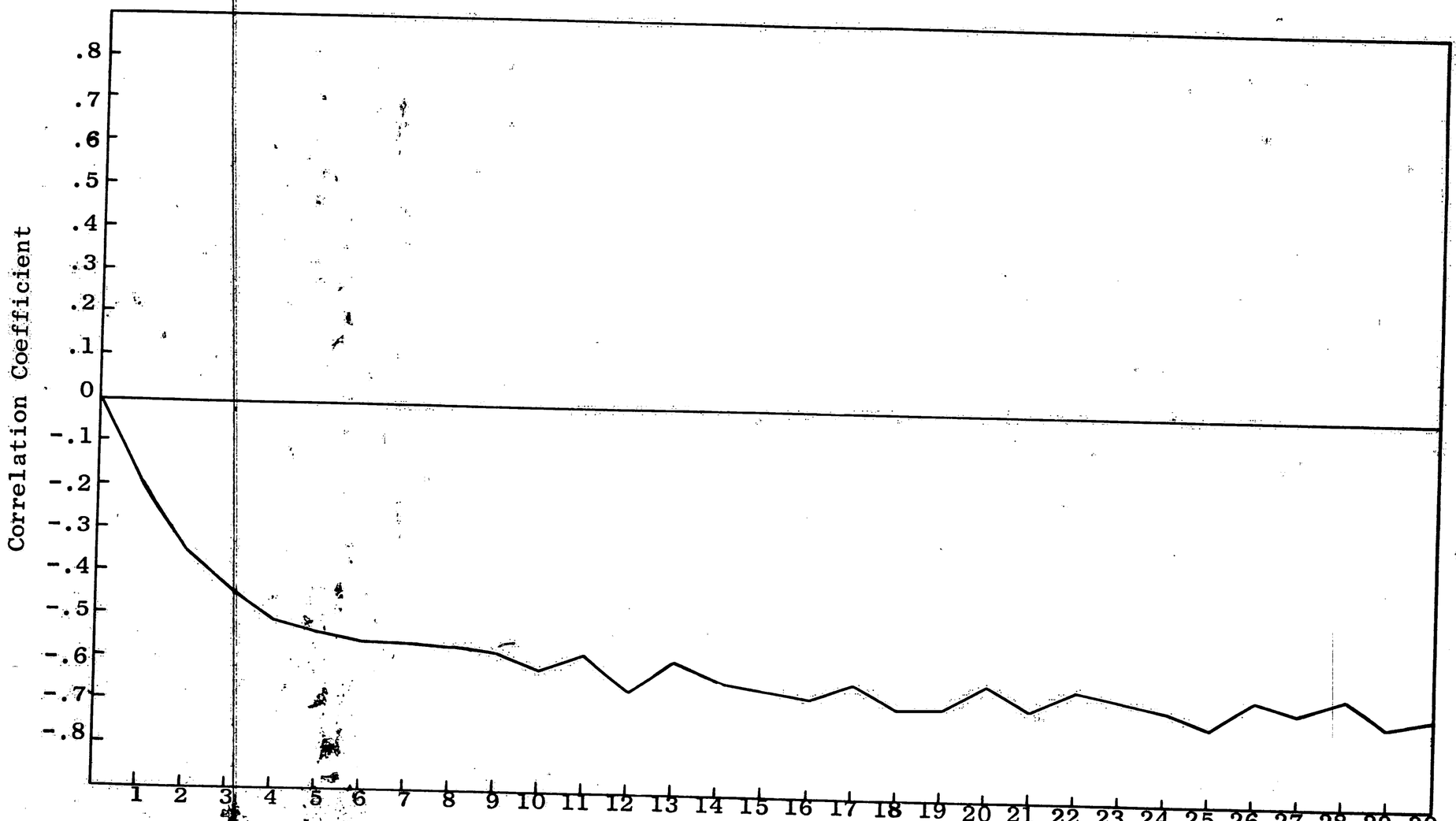
CORRELATION BETWEEN g_{cs} AND v_m FOR VARIOUS VALUES OF s
(s RANGES FROM .01 TO .30 IN INCREMENTS OF .01)

FIGURE A.19



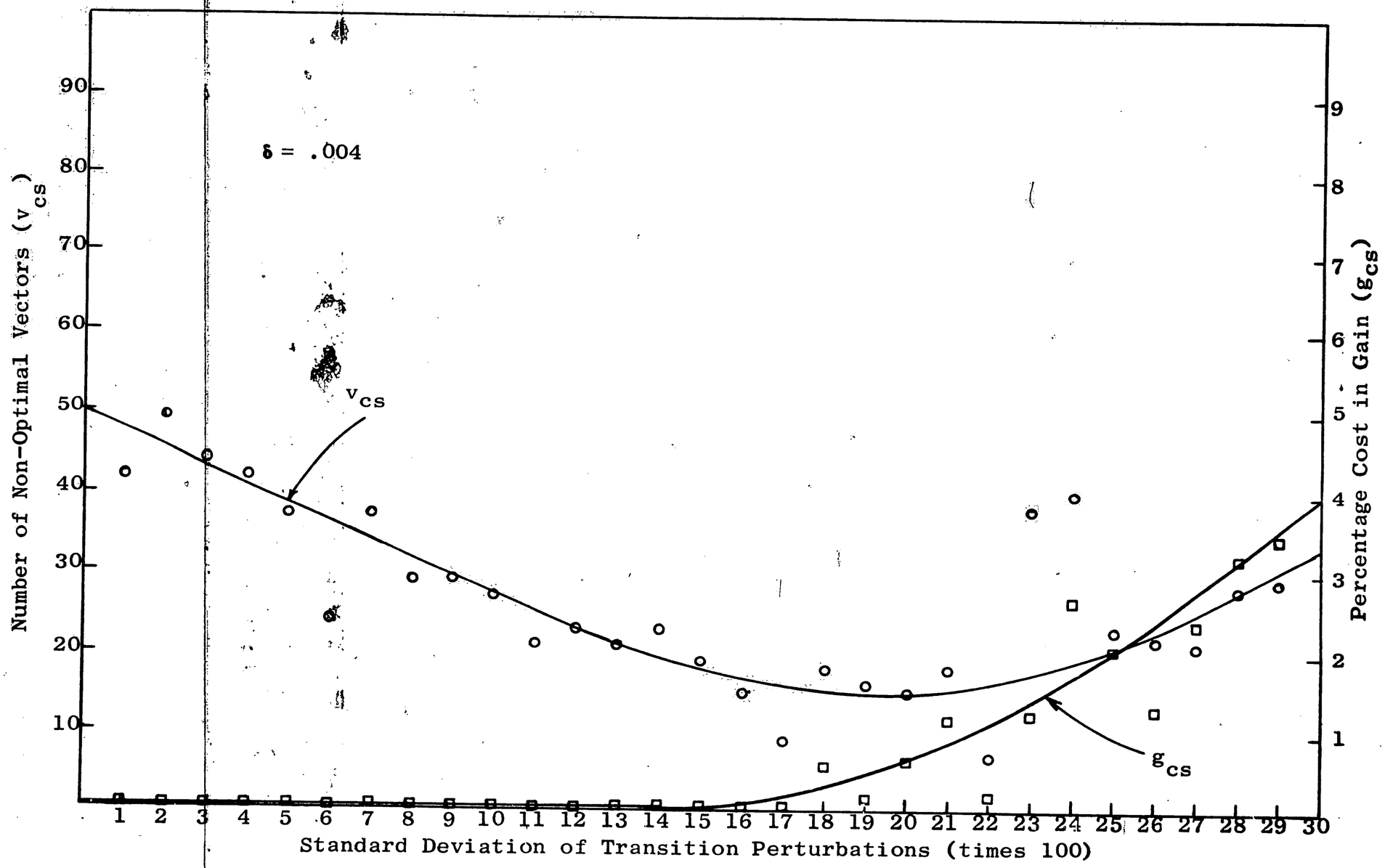
CORRELATION BETWEEN g_{cs} AND δ FOR VARIOUS VALUES OF s

FIGURE A.20

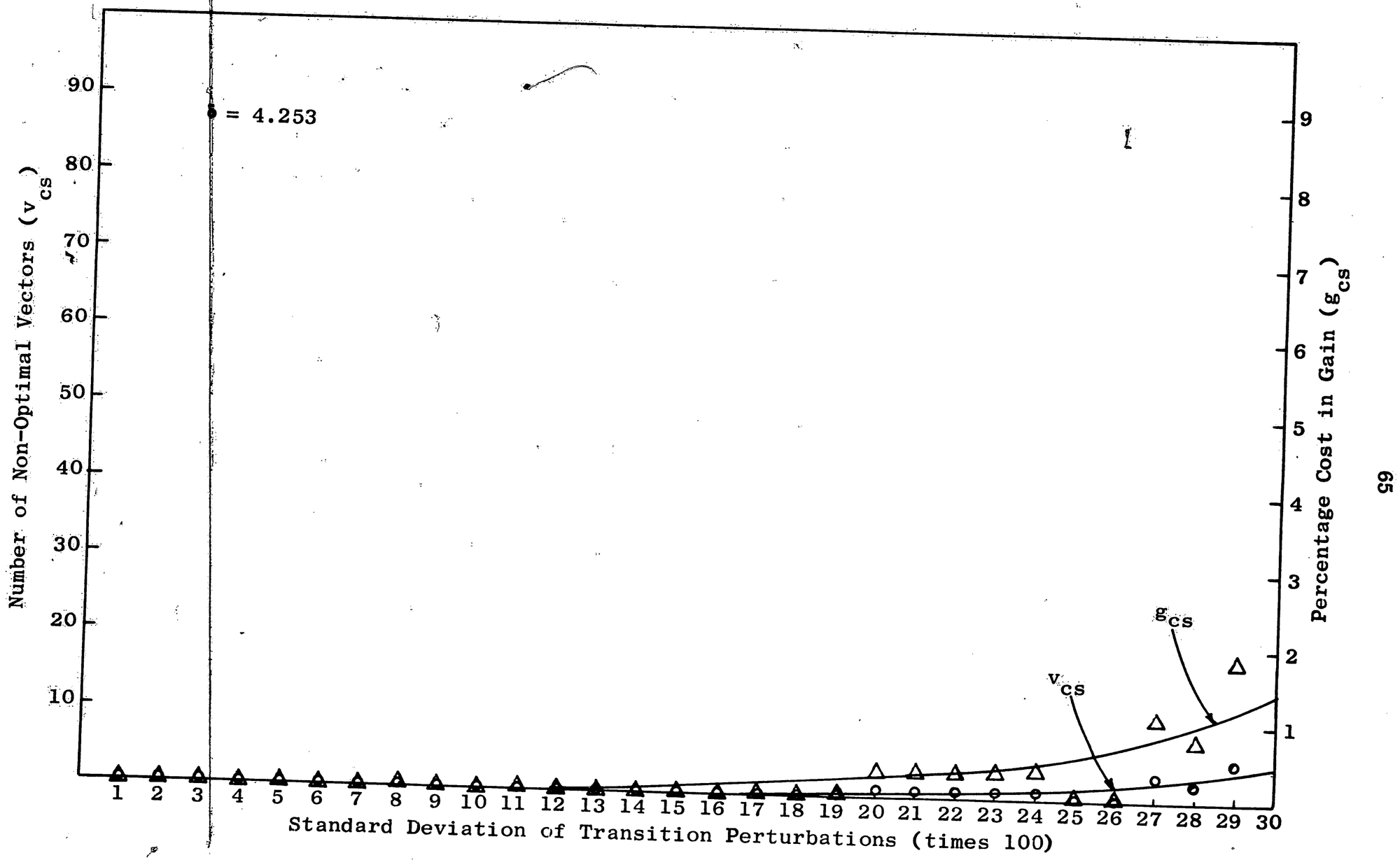


CORRELATION BETWEEN v_{cs} AND δ FOR VARIOUS VALUES OF s

FIGURE A.21



SENSITIVITY CURVES FOR δ = .004
FIGURE A.22



SENSITIVITY CURVES FOR $\delta = 4.253$
FIGURE A.23

BIBLIOGRAPHY

1. D'Epenoux, F., "A Probabilistic Production and Inventory System", Management Science, X, No. 1 (Oct. 1963), 98-109.

2. De Ghellinck, Guy T., and Gary D. Eppen, "Linear Programming Solutions for Separable Markovian Decision Problems", Management Science, XIII, No. 5 (Jan. 1967), 371-374.

3. Howard, Ronald A., Dynamic Programming and Markov Processes, The MIT Press, Massachusetts, 1964.

4. Karlin, Samuel, A First Course in Stochastic Processes, New York and London, Academic Press, 1966.

5. Kemeny, John G. and Laurie J. Snell, Finite Markov Chains, Princeton, N. J., D. Van Nostrand Company, Inc., 1960.

6. Kemeny, John G., Laurie J. Snell and Gerald L. Thompson, Introduction to Finite Mathematics, Englewood Cliffs, N. J., Prentice-Hall, Inc., 1961.

7. Kemeny, John G., Haxleton Merkil, Laurie J. Snell and Gerald L. Thompson, Finite Mathematical Structures, Englewood Cliffs, N. J., Prentice-Hall, Inc., 1959.

8. Manne, Alan S., "Linear Programming and Sequential Decisions", Management Science, VI, No. 3 (April 1960), 259-267.

9. Miller, David W. and Martin K. Starr, Inventory Control - Theory and Practice, Englewood Cliffs, N. J., Prentice-Hall, Inc., 1962.

VITA

PERSONAL HISTORY

Name: William L. Nutter

Place of Birth: Ramsey, West Virginia

Date of Birth: July 1, 1936

Parents: Navada I. Nutter
Lovell R. Nutter

Wife: Beryl A. Nutter

Children: Mark, Natalie and Bryce

EDUCATIONAL BACKGROUND

Ansted High School — Graduated 1954

West Virginia Institute of Technology
Bachelor of Science,
Electrical Engineering — Graduated 1961

Lehigh University
Master of Science,
Industrial Engineering — 1968

HONORS

Alpha Eta, Alpha Pi Mu

PROFESSIONAL EXPERIENCE

Western Electric Co., Inc.
Whippany, New Jersey
Development Engineer - Circuit Design and Reliability Engineering
July 1961 - January 1963

PROFESSIONAL EXPERIENCE (cont'd)

Western Electric Co., Inc.
White Sands Missile Range, New Mexico
Planning Engineer - Engineering Instruction
January 1963 - June 1966

Western Electric Co., Inc.
Princeton, New Jersey
Research Engineer - Lehigh Master's Program
June 1966 - Present